

```

1: ***      09:30 27/01/05
2: ****
3: ***      projet      2005
4: ***      pgm2005V2.s      version 68hc11
5: ***
6: ***      programme complet du contrôleur pH / conductivité
7: ****
8:
9:
10: ****
11: *      etiquettes
12: ****
13: debram      equ      $2000      * a modifier selon votre µc
14: pile      equ      $6fff      * idem et attention au vecteur d'it a ne pas ecraser
15:
16: vectit1      equ      $7fdb      * vecteur IT
17: toc1      equ      $1016
18: tmsk1      equ      $1022
19: tflg1      equ      $1023
20:
21: ****
22: * declaration des peripheriques d'entree/sortie paralleles
23: ****
24: porta      equ      $1000
25: portb      equ      $1004
26: portc      equ      $1003
27: porte      equ      $100a
28: ddrc      equ      $1007
29: pactl      equ      $1026
30:
31: din_mem      equ      $40      *
32: din_can      equ      $20      *      masques      *
33: memcan      equ      $10      *
34: hdata      equ      $08      *      port A      *
35: dout      equ      $02      *
36:
37:
38: cdepompe      equ      $80      *
39: selgam2      equ      $20      *      masques      *
40: selgam1      equ      $10      *
41: paramrap      equ      $04      *
42: e_aff      equ      $02      *      port C      *
43: rs_aff      equ      $01      *
44:
45:
46: t_clav      equ      $08      *      masque port E      *
47:
48:
49: ****
50: * vecteur d'interruption timer1
51: ****
52:      org      vectit1
53:      jmp      spinter1
54:
55: ****
56: * debut du prog en RAM
57: ****
58:      org      debram
59:      jmp      prog
60:
61: ****
62: *      texte a afficher
63: *      (table de caracteres)
64: ****
65:
66: * texte présentation projet 2005
67: ****
68: txtpres      fcc      "contrôleur pH"
69:      fcb      $00
70:      fcc      "conductivite"
71:      fcb      $00
72:
73: * texte température
74: ****
75: ttemp      fcb      $00
76:      fcc      " TEMP = "
77: ttempr      fcb      $20,$20,$20
78: ttemp2      fcb      "."

```

```

79: ttempr2 fcb      $20
80:          fcb      $20,$DF      *affichage symbole °C
81:          fcc      "C"
82:          fcb      $00
83:
84: * texte étalonnage température
85: *****
86: tetal1    fcc      "Etalonnage sonde"
87:          fcb      $00
88:          fcc      "temperature"
89:          fcb      $00
90: tetal2    fcc      "Etalonnage temp"
91:          fcb      $00
92:          fcb      $00
93:
94: * texte conductivité
95: *****
96: textco    fcc      " COND = "
97: tvco      fcb      $20,$20,$20,$20,$20
98:          fcb      $20,$E4      *affichage symbole µS
99:          fcc      "S"
100:         fcb      $00
101:         fcb      $00
102:
103: * texte étalonnage conductivité
104: *****
105: tetalc1    fcc      "Etalonnage sonde"
106:         fcb      $00
107:         fcc      "conductivite"
108:         fcb      $00
109: tetalc2    fcb      $00
110:         fcc      "Etalonnage coefc"
111:         fcb      $00
112: tetalc3    fcb      $00
113:         fcc      "Etalonnage coefT"
114:         fcb      $00
115:
116: * texte ph
117: *****
118: textph    fcc      " pH ="
119: tvph      fcb      $20,$20,$20
120: tph2      fcc      "."
121: tdph      fcb      $20
122:         fcb      $00
123:         fcb      $00
124:
125: * texte étalonnage ph
126: *****
127: tetalph1   fcc      "Etalonnage"
128:         fcb      $00
129:         fcc      "sonde pH"
130:         fcb      $00
131: tetalph2   fcc      "solution pH 7.00"
132:         fcb      $00
133:         fcc      "moins plus val"
134:         fcb      $00
135: tetalph3   fcc      "solution pH 9.00"
136:         fcb      $00
137:         fcc      "moins plus val"
138:         fcb      $00
139: tetalph4   fcc      "          SOL1"
140:         fcb      $00
141:         fcb      $00
142: tetalph5   fcc      "          SOL2"
143:         fcb      $00
144:         fcb      $00
145:
146: *****
147: *      variables d'affichage
148: *****
149:
150: nbaff      rmb      2      * sauvegarde du nb a afficher
151: adr_txt    rmb      2      * pointeur du texte au format nombre
152:
153: chif5      rmb      2      *
154: chif4      rmb      2      * equivalent bcd du nombre a afficher
155: chif3      rmb      2      * equivalent bcd du nombre a afficher
156: chif2      rmb      2      *

```

```

157: chif1          rmb      2      *
158:
159: achif5          rmb      1      *
160: achif4          rmb      1      * equivalent ascii du nombre a afficher
161: achif3          rmb      1      * equivalent ascii du nombre a afficher
162: achif2          rmb      1      *
163: achif1          rmb      1      *
164:
165: chif3D          rmb      2      * equivalent bcd du dixième a afficher
166: achif3D         rmb      1      * equivalent ascii du dixième a afficher
167: nbaffD          rmb      2      * sauvegarde du nb a afficher
168: adr_txD         rmb      2      * pointeur du texte au format nombre
169:
170:
171: *****
172: *      variables de gestion
173: *****
174:
175: compte          rmb      1
176: result          rmb      2
177: menu            rmb      1
178: moyenne         rmb      2
179:
180: *variables gestion timer
181: btimer          rmb      1
182: valth           rmb      2
183: valtl           rmb      2
184:
185: *variables gestion memoire
186: adeprom         rmb      1
187: cptpara         rmb      1
188: donnee          rmb      2
189:
190: * variables touches du clavier
191: seuil           rmb      1
192: phcond          rmb      1
193: etaltemp        rmb      1
194: etalph          rmb      1
195: etalcond        rmb      1
196: moins           rmb      1
197: plus            rmb      1
198: val             rmb      1
199:
200: *parametres de fonctionnement
201: pente           rmb      2      *ph
202: ph0             rmb      2      *ph
203: tempe           rmb      2      *ph
204: fctemp          rmb      2      *température
205: coeft           rmb      2      *conductivité
206: coefc           rmb      2      *conductivité
207: vseuil          rmb      2      *gestion seuil
208: znseuil         rmb      2      *gestion seuil
209: sseuil          rmb      2      *gestion seuil
210:
211: * variables pour le calcul de la température
212: signetemp       rmb      1      * - 0 temperature positive, $FF negative
213: vtemp           rmb      1      * Partie entiere de la temperature ( val absolue)
214: dtemp           rmb      1
215: restemp         rmb      2
216: tempmes         rmb      2
217:
218: * variables pour le calcul du ph
219: buff            rmb      2
220: quobuff1        rmb      2
221: respha          rmb      2
222: resphb          rmb      2
223: pha             rmb      1
224: phb             rmb      1
225: phmes           rmb      2
226: resph           rmb      2
227: vph             rmb      1
228: dph             rmb      1
229: solph           rmb      1
230: vsolph          rmb      2
231: dsolph          rmb      2
232: varph           rmb      2
233: phcor           rmb      2
234:

```

```

235: *variables pour le calcul de la conductivité
236: rescond      rmb      2
237: condm        rmb      2
238: condc        rmb      2
239: tcalc        rmb      2
240: gamme        rmb      1
241:
242: * Variables pour routine de multiplication
243: cal16x16      rmb      10      * Modifié par mul16_16
244:
245: * Variables pour routine de division
246: quoMSB        rmb      2      * Modifié par div32_16
247: quoLSB        rmb      2
248: reste_divM    rmb      2      * Modifié par div32_16 (4)
249: reste_divL    rmb      2
250: divisMSB      rmb      2      * Non modifié par div32_16 (8)
251: divisLSB      rmb      2
252: dividMSB      rmb      2      * Modifié par div32_16 (12)
253: dividLSB      rmb      2
254:
255:
256: *****
257: *****
258: ***
259: ***      programme principal
260: ***
261: *****
262: *****
263:
264: prog          lds      #pile      ; initialisation de l'adresse de pile
265:               ldy      #$1000    ; Y = adresse de base des B(R)SET ou B(R)CLR
266:               jsr      init      ; initialisation du systeme (ports,sorties,variables
267:               jsr      lectmem   ; charge les parametres de fonctionnement sauvegarde
268: retour        jsr      lec_clavier ; lecture et gestion du clavier
269:               jsr      ges_menu  ; gestion du menu
270:               jsr      lcant     ; lancement conversion de TempAna
271:               jsr      caltemp   ; calcul de la temperature
272:               jsr      afftemp   ; affichage de la temperature
273:               jsr      choixgam  ; choix gamme, conversion de CondAna, calcul de la c
274:               jsr      lcanp     ; lancement conversion PhAna
275:               jsr      cal_ph    ; calcul du pH
276:               jsr      affph     ; affichage du pH
277:               jsr      affcond   ; affichage de la conductivite
278:               jsr      ges_seuil ; gestion du seuil pH
279:               jsr      envph     ; envoie du pH par la boucle de courant
280:               bra      retour
281:
282:
283: *****
284: *****
285: ***
286: ***      liste des sous-programmes
287: ***
288: *****
289: *****
290:
291:
292: *****
293: ***
294: ***      SProgs d'initialisation: portC/sorties/variables/affichage
295: ***
296: *****
297:
298: *****
299: *      initialisation du systeme
300: *E:- S:- modif: a,ddrc,portB,gamme,seuil,phcond,etaltemp,etalph,etalcond,moins,plus,val
301: *****
302: init          jsr      initpor   ; initialisation du portC
303:               jsr      initsor   ; initialisation des sorties
304:               jsr      init_af   ; initialisation de l'afficheur
305:               jsr      clr_aff   ; effacement de l'afficheur
306:               jsr      affpres   ; affichage texte presentation du theme
307:               jsr      clr_aff   ; effacement de l'afficheur
308:               clr      gamme     ; initialisation variables
309:               clr      btimer
310:               clr      seuil
311:               clr      phcond
312:               clr      etaltemp

```

```

313:          clr      etalph
314:          clr      etalcond
315:          clr      moins
316:          clr      plus
317:          clr      val
318:          ldd      #250          ; tcalc = 25°C
319:          std      tcalc
320:          rts
321:
322: *****
323: *          initialisation du portC et du portA
324: *****
325: initpor          ldaa      #%11111111      * config du portc
326:                  staa      ddrc          * b0 a b7 en sortie
327:                  ldaa      #%00001000      * config PA3 en sortie
328:                  staa      pactl
329:                  rts
330:
331: *****
332: *          initialisation des sorties des ports
333: *****
334: initsor          clra
335:                  staa      portb          * code $00 sur le bus donnees de l'afficheur lcd
336:                  rts
337:
338:
339: *****
340: ***
341: ***          SProgs d'affichage de textes et de valeurs
342: ***
343: *****
344:
345: *****
346: *          initialisation de l'afficheur
347: * e: -          s: a          modif: a
348: *****
349: init_af          ldaa      #%00111000
350:                  jsr      instruc
351:                  ldaa      #%00001100
352:                  jsr      instruc
353:                  ldaa      #%000000110
354:                  jsr      instruc
355:                  rts
356:
357: *****
358: *          effacement complet de l'afficheur
359: * e: -          s: a          modif: a
360: *****
361: clr_aff          ldaa      #%000000001
362:                  jsr      instruc
363:                  rts
364:
365: *****
366: *          envoi & validation d'une instruction
367: * e: a          s: -          modif: a
368: *****
369: instruc          staa      portb
370:                  bclr      portc,y e_aff
371:                  bclr      portc,y rs_aff
372:                  bset      portc,y e_aff
373:                  bclr      portc,y e_aff
374:                  jsr      tp10ms
375:                  rts
376:
377: *****
378: *          envoi & validation d'un caractere
379: * e: a          s: -          modif: a
380: *****
381: caract          staa      portb
382:                  bclr      portc,y e_aff
383:                  bclr      portc,y rs_aff
384:                  bset      portc,y rs_aff
385:                  bset      portc,y e_aff
386:                  bclr      portc,y e_aff
387:                  jsr      tp10ms
388:                  rts
389:
390: *****

```

```

391: *      envoi d'un texte sur deux lignes
392: * e: x      s: a      modif: a,x
393: *****
394: aff_txt      ldaa      #%10000000
395:              jsr      instruc
396: lg1          ldaa      0,x
397:              beq      txt_st
398:              jsr      caract
399:              inx
400:              bra      lg1
401: txt_st      ldaa      #%11000000
402:              jsr      instruc
403:              inx
404: lg2          ldaa      0,x
405:              beq      txt_fin
406:              jsr      caract
407:              inx
408:              bra      lg2
409: txt_fin rts
410:
411: *****
412: * affichage texte presentation theme 2005
413: *****
414: affpres      ldx      #txtpres      ; charge dans X l'adresse de debut du texte à affich
415:              jsr      aff_txt      ; affiche le texte
416:              jsr      tp1s      ; attente 2s
417:              jsr      tp1s
418:              rts
419:
420: *****
421: * Sprogs affichage de la temperature et du pH
422: *****
423:
424: *****
425: *      SProg affichage de la température
426: * e: vtemp, dtemp      s: x,d      mofif: x,d
427: *****
428: afftemp      clra
429:              ldab      vtemp
430:              ldx      #ttempr      ; adresse debut texte temperature
431:              jsr      aff_nb      ; preparation affichage valeur temperature
432:              clra
433:              ldab      dtemp
434:              ldx      #ttempr2
435:              jsr      aff_nbD
436:              ldx      #ttemp
437:              jsr      aff_txt      ; affichage temperature
438:              rts
439:
440: *****
441: *      SProg affichage de la valeur du ph
442: * e: vph, dph      s: d,x      mofif: d,x
443: *****
444: affph        clra
445:              ldab      vph
446:              ldx      #tvph      ; adresse debut texte pH
447:              jsr      aff_nb      ; preparation affichage valeur pH
448:              clra
449:              ldab      dph
450:              ldx      #tdph
451:              jsr      aff_nbD
452:              ldx      #textph
453:              tst      phcond
454:              bne      finafph
455:              jsr      aff_txt      ; affichage pH
456: finafph rts
457:
458: *****
459: *      SProg affichage pha et phb pour l'étalonnage du pH
460: * e: solph      s: -      mofif: d,x
461: *****
462: affetph      clra
463:              ldab      solph
464:              ldx      #10
465:              idiv
466:              stx      vsolph
467:              std      dsolph
468:              ldd      vsolph

```

```

469:         ldx      #tvph
470:         jsr      aff_nb
471:         ldd      dsolph
472:         ldx      #tdph
473:         jsr      aff_nbD
474:         ldx      #textph
475:         jsr      aff_txt
476:         rts
477:
478: *****
479: * affiche un nombre sur 3 chiffres (centaine,dizaine,unité: température et pH)
480: * e: d,x      s: d,x,nbaff,adr_txt      modif: d,x
481: *****
482: aff_nb      std      nbaff      ; sauvegarde du nombre a afficher
483:             stx      adr_txt    ; sauvegarde du pointeur du nombre a afficher
484:             jsr      hexadec    ; conversion hexa/decimale (d)
485:             jsr      inbasci    ; init des nombres au format ascii
486:             jsr      decasci    ; conversion decimale/ascii (tabdec=>tabasci)
487:             jsr      nb_text
488:             rts
489:
490: *****
491: * sous programme de conversion hexa/decimale (température et pH)
492: * e: d      s: chif3..chif1      modif: d,x
493: *****
494: hexadec      ldx      #100
495:             idiv
496:             stx      chif3
497:             ldx      #10
498:             idiv
499:             stx      chif2
500:             std      chif1
501:             rts
502:
503: *****
504: * initialisation des nombres/ascii (température et pH)
505: * e: -      s: achif3..achif1      modif: a
506: *****
507: inbasci      ldaa     #$20
508:             staa     achif3
509:             staa     achif2
510:             staa     achif1
511:             rts
512:
513: *****
514: * procedure de conversion decimale/ascii (température et pH)
515: * e: chif3..chif1      s: achif3..achif1      modif: d
516: *****
517: decasci      ldd      chif3      ; msb=0 et lsb=nb_bcd
518:             beq      deca1
519:             addb     #$30
520:             stab     achif3
521: deca1      aba      ; sauvegarde des poids additionner dans A
522:             clrb
523:             addd     chif2
524:             beq      deca2
525:             addb     #$30
526:             stab     achif2
527: deca2      ldd      chif1
528:             addb     #$30
529:             stab     achif1
530:             rts
531:
532: *****
533: * procedure de transfert valeur => texte (température et pH)
534: * e: -      s: -      modif: x,d
535: *****
536: nb_text      ldx      adr_txt
537:             ldaa     achif3
538:             staa     0,x
539:             ldaa     achif2
540:             staa     1,x
541:             ldaa     achif1
542:             staa     2,x
543:             rts
544:
545: *****
546: *         affiche le dixième de la température et du pH

```

```

547: * e: d,x      s: d,x,nbaff,adr_txt      modif: d,x
548: *****
549: aff_nbD      std      nbaffD      ; sauvegarde du nombre a afficher
550:      stx      adr_txtD      ; sauvegarde du pointeur du nombre a afficher
551:      std      chif3D
552:      jsr      inbasCD      ; init des nombres au format ascii
553:      jsr      decascD      ; conversion decimale/ascii (tabdec=>tabasci)
554:      jsr      nb_texD
555:      rts
556:
557: *****
558: * initialisation des nombres/ascii (dizième température et pH)
559: * e: -      s: achif3D      modif: a
560: *****
561: inbasCD      ldaa      #$30
562:      staa      achif3D
563:      rts
564:
565: *****
566: * procedure de conversion decimale/ascii (dizième température et pH)
567: * e: chif3D      s: achif3D      modif: d
568: *****
569: decascD      ldd      chif3D      ; msb=0 et lsb=nb_bcd
570:      beq      deca1D
571:      addb      #$30
572:      stab      achif3D
573: deca1D      rts
574:
575: *****
576: * procedure de transfert valeur => texte (dizième température et pH)
577: * e: adr_txtD, achif3D      s: -      modif: x,a
578: *****
579: nb_texD      ldx      adr_txtD
580:      ldaa      achif3D
581:      staa      0,x
582:      rts
583:
584:
585: *****
586: *      Sprogs affichage de la conductivité
587: *****
588:
589: *****
590: *      SProg affichage de la valeur de la conductivité
591: * e: condC      s: d,x      mofif: d,x
592: *****
593: affcond      ldd      condC
594:      ldx      #tvco      * adresse début texte conductivité
595:      jsr      aff_nbC      * préparation affichage valeur conductivité
596:      ldx      #textco
597:      tst      phcond      * test si autorisation affichage conductivité
598:      beq      finaffC
599:      jsr      aff_txt      * affichage valeur et texte conductivité
600: finaffC      rts
601:
602: *****
603: *      SProg affichage étalonnage conductivité
604: * e: condC      s: d,x      mofif: d,x
605: *****
606: affetalC      ldd      condC
607:      ldx      #tvco      * adresse début texte conductivité
608:      jsr      aff_nbC      * préparation affichage valeur conductivité
609:      ldx      #textco
610:      jsr      aff_txt      * affichage valeur et texte conductivité
611:      rts
612:
613: *****
614: *      affiche la valeur de la conductivité
615: * e: d,x      s: -      modif: d,x
616: *****
617: aff_nbC      std      nbaff      ; sauvegarde du nombre a afficher
618:      stx      adr_txt      ; sauvegarde du pointeur du nombre a afficher
619:      jsr      hexadC      ; conversion hexa/decimale (d)
620:      jsr      inbasC      ; init des nombres au format ascii
621:      jsr      decascC      ; conversion decimale/ascii (tabdec=>tabasci)
622:      jsr      nb_texC
623:      rts
624:

```

```

625: *****
626: * sous programme de conversion hexa/decimale (conductivité)
627: * e: d s: chif5..chif1 modif: d,x
628: *****
629: hexadC      ldx      #10000
630:             idiv
631:             stx      chif5
632:             ldx      #1000
633:             idiv
634:             stx      chif4
635:             ldx      #100
636:             idiv
637:             stx      chif3
638:             ldx      #10
639:             idiv
640:             stx      chif2
641:             std      chif1
642:             rts
643:
644: *****
645: * initialisation des nombres/ascii (conductivité)
646: *****
647: inbasC      ldaa     #$20
648:             staa     achif5
649:             staa     achif4
650:             staa     achif3
651:             staa     achif2
652:             staa     achif1
653:             rts
654:
655: *****
656: * procedure de conversion decimale/ascii (conductivité)
657: * e: - s: achif5..achif1 modif: d,x
658: *****
659: decascC     ldd      chif5 ; msb=0 et lsb=nb_bcd
660:             beq      deca1c
661:             addb     #$30
662:             stab     achif5
663: deca1c      aba
664:             clrb
665:             addd     chif4
666:             beq      deca2c
667:             addb     #$30
668:             stab     achif4
669: deca2c      aba
670:             clrb
671:             addd     chif3
672:             beq      deca3c
673:             addb     #$30
674:             stab     achif3
675: deca3c      aba
676:             clrb
677:             addd     chif2
678:             beq      deca4c
679:             addb     #$30
680:             stab     achif2
681: deca4c      ldd      chif1
682:             addb     #$30
683:             stab     achif1
684:             rts
685:
686: *****
687: * procedure de transfert valeur => texte (conductivité)
688: * e: x s: - modif: x,d
689: *****
690: nb_textC    ldx      adr_txt
691:             ldaa     achif5
692:             staa     0,x
693:             ldaa     achif4
694:             staa     1,x
695:             ldaa     achif3
696:             staa     2,x
697:             ldaa     achif2
698:             staa     3,x
699:             ldaa     achif1
700:             staa     4,x
701:             rts
702:

```

```

703:
704: *****
705: *      SProgs étalonnage des sondes température, pH et conductivité
706: *****
707:
708: *****
709: *      SProg étalonnage de la sonde température
710: * e: -      s: -      modif: fctemp
711: *****
712: etalon_temp      jsr      clr_aff
713:                  ldx      #tetalt1      ; affichage message
714:                  jsr      aff_txt
715:                  jsr      tp1s
716:                  jsr      tp1s
717:                  jsr      clr_aff
718:                  ldx      #tetalt2
719:                  jsr      aff_txt
720: s0_etalon_temp    jsr      lcant      ; lancement conversion TempAna
721:                  jsr      caltemp     ; calcul de la temperature
722:                  jsr      afftemp     ; affichage de la temperature
723:                  jsr      lec_clavier ; lecture du clavier
724:                  tst      plus        ; test appui touche plus
725:                  beq      s1_etalon_temp ; saut à l'étiquette si pas appui sur touche plus
726:                  clr      plus
727:                  ldx      fctemp
728:                  inx
729:                  stx      fctemp      ; incremente fctemp si appui sur BP plus
730:                  jmp      s0_etalon_temp
731: s1_etalon_temp    tst      moins      ; test appui touche moins
732:                  beq      s2_etalon_temp ; saut à l'étiquette si pas appui sur touche moins
733:                  clr      moins
734:                  ldx      fctemp
735:                  dex
736:                  stx      fctemp      ; decremente fctemp si appui sur BP moins
737:                  jmp      s0_etalon_temp
738: s2_etalon_temp    tst      val        ; test appui touche val
739:                  beq      s0_etalon_temp ; fin etalonnage si appui sur BP val sinon retour à
740:                  clr      val
741:                  jsr      clr_aff
742:                  rts
743:
744: *****
745: *      SProg étalonnage sonde pH
746: * e: -      s: pente,ph0,tempe,solph      modif: y,x,d,resph,tempmes,respha,resphb,pha,phb
747: *****
748: etalon_ph      jsr      clr_aff
749:                  ldx      #tetalph1
750:                  jsr      aff_txt      ; affichage message pendant 2s
751:                  jsr      tp1s
752:                  jsr      tp1s
753:                  jsr      clr_aff
754:                  ldx      #tetalph2
755:                  jsr      aff_txt
756:                  jsr      tp1s
757:                  jsr      tp1s
758:                  jsr      clr_aff
759:                  ldx      #tetalph4
760:                  jsr      aff_txt      ; affichage message
761:                  ldaa     #70
762:                  staa     solph      ; solph=70
763: s0_etal_ph      jsr      affetph      ; affichage valeur solution etalon
764:                  jsr      lcant      ; lancement conversion TempAna
765:                  jsr      caltemp     ; calcul de la temperature
766:                  jsr      afftemp     ; affichage temperature
767:                  jsr      lec_clavier ; lecture clavier
768:                  tst      etalph
769:                  beq      s1_etal_ph
770:                  jmp      fin_etal_ph ; fin etalonnage si appui BP etalph
771: s1_etal_ph      tst      plus
772:                  beq      s2_etal_ph
773:                  clr      plus
774:                  inc      solph      ; incremente solph si appui sur BP plus
775:                  jsr      affetph      ; affichage valeur solution etalon
776: s2_etal_ph      tst      moins
777:                  beq      s3_etal_ph
778:                  clr      moins
779:                  dec      solph      ; decremente solph si appui sur BP moins
780:                  jsr      affetph      ; affichage valeur solution etalon

```

```

781: s3_etal_ph      tst      val
782:                  beq      s0_etal_ph
783:                  clr      val
784:                  ldaa     solph
785:                  staa     phb          ; phb=solph si appui sur BP val
786:                  jsr      lcanp        ; lancement conversion PhAna
787:                  ldd      resph
788:                  std      resphb       ; resphb=resph
789:                  ldd      tempmes
790:                  std      tempe        ; tempe=tempmes
791:
792:                  jsr      clr_aff
793:                  ldx      #tetalph3
794:                  jsr      aff_txt      ; affichage message pendant 2s
795:                  jsr      tp1s
796:                  jsr      tp1s
797:                  jsr      clr_aff
798:                  ldx      #tetalph5
799:                  jsr      aff_txt      ; affichage message
800:                  jsr      afftemp
801:                  ldaa     #90
802:                  staa     solph         ; solph=90
803:                  jsr      affetph       ; affichage valeur solution etalon
804: s4_etal_ph      jsr      lcant         ; lancement conversion TempAna
805:                  jsr      caltemp      ; calcul de la temperature
806:                  jsr      afftemp      ; affichage temperature
807:                  jsr      lec_clavier  ; lecture clavier
808:                  tst      etalph
809:                  beq      s5_etal_ph
810:                  jmp      fin_etal_ph  ; fin etalonnage si appui BP etalph
811: s5_etal_ph      tst      plus
812:                  beq      s6_etal_ph
813:                  clr      plus
814:                  inc      solph         ; incremente solph si appui sur BP plus
815:                  jsr      affetph       ; affichage valeur solution etalon
816: s6_etal_ph      tst      moins
817:                  beq      s7_etal_ph
818:                  clr      moins
819:                  dec      solph         ; decremente solph si appui sur BP moins
820:                  jsr      affetph       ; affichage valeur solution etalon
821: s7_etal_ph      tst      val
822:                  beq      s4_etal_ph
823:                  clr      val
824:                  ldaa     solph
825:                  staa     pha          ; pha=solph si appui sur BP val
826:                  jsr      lcanp        ; lancement conversion PhAna
827:                  ldd      resph
828:                  std      respha       ; respha=resph
829:                  jsr      clr_aff
830:
831:                  pshy
832:                  clr      buff          ; initialisation du signe
833:                  ldd      resphb       ; Calcul de la pente
834:                  subd     respha       ; resphb-respha
835:                  bpl      s8_etal_ph
836:                  coma
837:                  comb
838:                  addd     #1
839:                  com      buff
840: s8_etal_ph      ldx      #100
841:                  jsr      mul16_16     ; (resphb-respha)*100
842:                  stx      dividMSB
843:                  sty      dividLSB
844:                  clra
845:                  ldab     pha
846:                  subb     phb          ; pha-phb
847:                  bne     s9_etal_ph
848:                  ldd      #100        ; Division par 0, 100 dans pente
849:                  std      pente
850:                  bra      cal_ph0
851: s9_etal_ph      bpl      s10_etal_ph
852:                  negb
853:                  com      buff
854: s10_etal_ph     std      divisLSB
855:                  ldx      #quoMSB
856:                  jsr      div32_16
857:                  ldd      quoLSB
858:                  tst      buff

```

```

859:          beq      s11_etal_ph
860:          coma
861:          comb
862:          addd     #1
863: s11_etal_ph std      pente           ; pente=(pha-phb)/((resphb-respha)*100)
864:
865: cal_ph0 clr      buff              ; calcul de ph0
866:          clra
867:          ldab     pha
868:          subb     phb              ; pha-phb
869:          bpl      s0_cal_ph0
870:          coma
871:          comb
872:          addd     #1
873:          com      buff
874: s0_cal_ph0 ldx     respha
875:          jsr      mul16_16         ; respha*(pha-phb)
876:          stx      dividMSB
877:          sty      dividLSB
878:
879:          ldd      resphb           ; resphb-respha
880:          subd     respha
881:          bne      s1_cal_ph0
882:          ldd      #70              ; Division par 0, on met 7,0 pour le Ph0
883:          std      ph0
884:          rts
885: s1_cal_ph0 bpl      s2_cal_ph0
886:          coma
887:          comb
888:          addd     #1
889:          com      buff
890: s2_cal_ph0 std      divisLSB
891:          ldx      #quoMSB
892:          jsr      div32_16         ; (resphb-respha)/(respha*(pha-phb))
893:          ldd      quoLSB
894:          tst      buff
895:          beq      s3_cal_ph0
896:          coma
897:          comb
898:          addd     #1
899: s3_cal_ph0 ldab     pha
900:          clra
901:          addd     quoLSB
902:          std      ph0             ; ph0=pha+(resphb-respha)/(respha*(pha-phb))
903:          puly
904: fin_etal_ph clr     etalph
905:          jsr      clr_aff
906:          rts
907:
908: *****
909: *          SProg étalonnage de la conductivité
910: * e: - s: -   modif: x,d,tcalc,coeft,coefc
911: *****
912: etalon_cond jsr      clr_aff
913:          clr      gamme
914:          ldx      #tetalc1
915:          jsr      aff_txt          ; affichage message pendant 2s
916:          jsr      tp1s
917:          jsr      tp1s
918:          jsr      clr_aff
919:          ldx      #tetalc2
920:          jsr      aff_txt          ; affichage message pendant 2s
921:          jsr      tp1s
922:          jsr      tp1s
923:          jsr      clr_aff
924: s0_etalon_cond jsr     lcant         ; lancement conversion TempAna
925:          jsr      caltemp          ; calcul de la temperature
926:          jsr      afftemp          ; affichage temperature
927:          ldd      tempmes
928:          std      tcalc            ; tcalc=tempmes
929:          jsr      choixgam         ; conversion CondAna, choix gamme, calcul conductivi
930:          jsr      affetalc         ; affichage conductivite
931:          jsr      lec_clavier      ; lecture clavier
932:          tst      plus
933:          beq      s1_etalon_cond
934:          clr      plus
935:          ldx      coefc
936:          inx

```

```

937:                                stx    coefc            ; incremente coefc si appui sur BP plus
938: s1_etalon_cond                tst    moins
939:                                beq    s2_etalon_cond
940:                                clr    moins
941:                                ldx    coefc
942:                                dex
943:                                stx    coefc            ; decremente coefc si appui sur BP moins
944: s2_etalon_cond                tst    val
945:                                beq    s3_etalon_cond
946:                                clr    val
947:                                jmp    s4_etalon_cond
948: s3_etalon_cond                tst    etalcond
949:                                beq    s0_etalon_cond
950:                                clr    etalcond
951:                                jmp    fin_etalon_c      ; fin etalonnage si appui sur BP etalcond
952:
953: s4_etalon_cond                jsr    clr_aff
954:                                ldx    #tetalc3
955:                                jsr    aff_txt            ; affichage message pendant 2s
956:                                jsr    tp1s
957:                                jsr    tp1s
958:                                jsr    clr_aff
959: s5_etalon_cond                jsr    lcant              ; lancement conversion TempAna
960:                                jsr    caltemp           ; calcul de la temperature
961:                                jsr    afftemp           ; affichage temperature
962:                                ldd    #250
963:                                std    tcalc            ; tcalc = 25°C
964:                                jsr    choixgam          ; conversion CondAna, choix gamme, calcul conductivi
965:                                jsr    affetalc         ; affichage conductivite
966:                                jsr    lec_clavier       ; lecture clavier
967:                                tst    plus
968:                                beq    s6_etalon_cond
969:                                clr    plus
970:                                ldx    coeft
971:                                inx
972:                                stx    coeft            ; incremente coeft si appui sur BP plus
973: s6_etalon_cond                tst    moins
974:                                beq    s7_etalon_cond
975:                                clr    moins
976:                                ldx    coeft
977:                                dex
978:                                stx    coeft            ; decremente coeft si appui sur BP moins
979: s7_etalon_cond                tst    val
980:                                beq    s8_etalon_cond
981:                                clr    val
982:                                jmp    fin_etalon_c      ; fin etalonnage si appui sur BP val
983: s8_etalon_cond                tst    etalcond          ; fin etalonnage si appui sur BP etalcond
984:                                beq    s5_etalon_cond
985:                                clr    etalcond
986: fin_etalon_c                  jsr    clr_aff
987:                                ldd    #250
988:                                std    tcalc            ; tcalc = 25°C
989:                                rts
990:
991:
992: *****
993: *          SProgs calcul temperature, pH et conductivite
994: *****
995:
996: *****
997: *          SProg calcul de la temperature
998: * e: fctemp,restemp          s: vtemp,dtemp,tempmes   modif: y,x,d
999: *****
1000: caltemp                        pshy
1001:                                clr    signetemp
1002:                                ldd    fctemp
1003:                                addd   restemp
1004:                                subd   #120              ; on effectue (Restemp -120 + Fctemp)
1005:                                bpl    s0_caltemp
1006:                                coma
1007:                                comb
1008:                                addd   #1                ; multiplication par -1
1009:                                com    signetemp
1010: s0_caltemp                     ldx    #2500            ; on multiplie par 2500 pour
1011:                                jsr    mul16_16          ; travailler en 1/20 de degre
1012:                                xgdy                    ; Resultatsur 24 bits
1013:                                tab                    ; division par 4096
1014:                                pshb

```

```

1015:      xgdx
1016:      tba
1017:      pulb
1018:      lsrđ
1019:      lsrđ
1020:      lsrđ
1021:      lsrđ      ; Fin division
1022:      tst      signetemp      ; test si resultat negatif
1023:      beq      s1_caltemp
1024:      coma
1025:      comb
1026:      addd      #1      ; multiplication par -1
1027:      clr      signetemp
1028: s1_caltemp      subd      #250      ; on soustrait 12,5 degre soit 250
1029:      bpl      s2_caltemp      ; test si resultat negatif
1030:      coma
1031:      comb
1032:      addd      #1      ; multiplication par -1 et memorisation
1033:      com      signetemp      ; signe negatif
1034: s2_caltemp      clr      vtemp
1035:      ldx      #20      ; division par 10 de la valeur absolue
1036:      idiv      ; de la temperature
1037:      lsrđ      ; division par 2 et arrondi à 0,05 degre
1038:      bcc      s3_caltemp
1039:      incb
1040:      cmpb      #10
1041:      blo      s3_caltemp
1042:      inc      vtemp
1043:      clrb
1044: s3_caltemp      stab      dtemp      ; Partie decimale 0 à 9
1045:      xgdx
1046:      addb      vtemp      ; partie entiere 0 à 255.
1047:      stab      vtemp
1048:      ldaa      vtemp      ; calcul de tempmes
1049:      ldab      #10
1050:      mul
1051:      std      tempmes
1052:      clra
1053:      ldab      dtemp
1054:      addd      tempmes
1055:      std      tempmes
1056: f_caltemp      puly
1057:      rts
1058:
1059: *****
1060: *      SProg calcul du pH
1061: * e: pente, resph, ph0, vtemp, tempe      s: phmes, vph, dph, phcor      modif: y, x, d
1062: *****
1063: cal_ph      pshy
1064:      clr      buff
1065:      ldd      resph
1066:      ldx      #100
1067:      jsr      mul16_16      ; resph * 100
1068:      stx      dividMSB
1069:      sty      dividLSB
1070:      ldd      pente
1071:      bpl      s0_cal_ph
1072:      coma
1073:      comb
1074:      addd      #1
1075:      com      buff
1076: s0_cal_ph      std      divisLSB
1077:      ldx      #quoMSB
1078:      jsr      div32_16      ; pente / (resph*100)
1079:      ldd      quoLSB
1080:      tst      buff
1081:      beq      s1_cal_ph
1082:      coma
1083:      comb
1084:      addd      #1
1085:      std      quoLSB
1086: s1_cal_ph      ldd      ph0
1087:      subd      quoLSB
1088:      std      phmes      ; phmes = ph0 - (pente / (resph*100))
1089:      ldd      resph
1090:      ldx      #25
1091:      jsr      mul16_16      ; resph * 25
1092:      sty      dividLSB

```

```

1093:      stx      dividMSB
1094:      clra
1095:      ldd      tempmes
1096:      addd     #2730          ; tempmes + 2730
1097:      std      divisLSB
1098:      ldx      #quoMSB
1099:      jsr      div32_16
1100:      ldd      quoLSB
1101:      std      quobuff1      ; quobuff1=(tempmes+2730)/(resph*500)
1102:      ldd      resph
1103:      ldx      #25
1104:      jsr      mul16_16      ; resph * 25
1105:      sty      dividLSB
1106:      stx      dividMSB
1107:      clra
1108:      ldd      tempe
1109:      addd     #2730          ; tempe + 2730
1110:      std      divisLSB
1111:      ldx      #quoMSB
1112:      jsr      div32_16
1113:      ldd      quoLSB
1114:      subd     quobuff1      ; (tempmes+2730)/(resph*500)-quobuff1
1115:      bpl      s2_cal_ph
1116:      coma
1117:      comb
1118:      addd     #1
1119:      std      buff
1120:      ldd      phmes
1121:      subd     buff
1122:      bra      s3_cal_ph
1123: s2_cal_ph addd     phmes
1124: s3_cal_ph std      phcor      ; phcor=phmes+((tempmes+2730)/(resph*500)-quobuff1)
1125:      ldx      #10
1126:      idiv
1127:      xgdx
1128:      stab     vph          ; dizaine et unite pH
1129:      xgdx
1130:      stab     dph          ; dixième pH
1131:      puly
1132:      rts
1133:
1134: *****
1135: *      SProgs mesure et calcul de la conductivité
1136: * e: tcalc,tempmes,coeft,coefc s: condc      modif: y,x,d
1137: *****
1138:
1139: *****
1140: * sous prog CHOIXGAM (choix de la gamme en fonction de la valeur de la conductivité)
1141: *****
1142: choixgam      tst      gamme      *teste le choix de la gamme précédente
1143:               bne      chgam2
1144: chgam1        clr      gamme
1145:               bset     portc,y selgam1 *sélection gamme1 (1000 à 10000 µS)
1146:               bclr     portc,y selgam2
1147:               jsr      tp100ms      *attente 100ms
1148:               jsr      lcanc      *lancement conversion de la tension condana
1149:               jsr      mcondg1     *mesure conductivité avec la gamme 1
1150:               jsr      calcond     *calcul de la conductivité
1151:               ldd      condc      *si conductivité < 1000µS alors choix
1152:               cpd      #1000      *de la gamme 2
1153:               bmi      chgam2
1154:               jmp      finchgam
1155: chgam2        bset     portc,y selgam2 *sélection gamme2 (100 à 1000 µS)
1156:               bclr     portc,y selgam1
1157:               ldaa     #1
1158:               staa     gamme
1159:               jsr      tp100ms      *attente 100ms
1160:               jsr      lcanc      *lancement conversion de la tension condana
1161:               jsr      mcondg2     *mesure conductivité avec la gamme 2
1162:               jsr      calcond     *calcul de la conductivité
1163:               ldd      condc      *si conductivité > 1000µS alors choix
1164:               cpd      #1000      *de la gamme 1
1165:               bhi      chgam1
1166: finchgam      rts
1167:
1168:
1169: *****
1170: *sous prog CALCOND (calcul de la conductivité corrigée en fonction de la température)

```

```

1171: *      condc = condm * ( 1 + (( coeft / 10000 ) * ( tcalc - tempmes )))
1172: *****
1173: calcond      pshy
1174:             clr      buff      * Signe de l'operation
1175:             ldd      tcalc      * chargement valeur de tcalc
1176:             subd     tempmes     * soustraction tcalc - tempmes
1177:             bpl      s0_calcond  * test si différence négative
1178:             coma      * complément à 2 si différence négative
1179:             comb
1180:             addd      #1
1181:
1182:             com      buff
1183: s0_calcond   ldx      coeft      * coeft: coefficient de température de la sonde
1184:             jsr      mul16_16    * multiplication par coeft
1185:             sty      varph
1186:             ldd      varph
1187:             cpd      #10000
1188:             bls      s1_calcond  * limitation de la correction à +/-1
1189:             ldd      #1000
1190: s1_calcond   ldx      condm
1191:             jsr      mul16_16    * multiplication par condm
1192:             stx      dividMSB    * début division par 1000
1193:             sty      dividLSB
1194:             ldd      #10000
1195:             std      divisLSB
1196:             ldd      #0
1197:             std      divisMSB
1198:             ldx      #quoMSB     * adresses des étiquettes pour la division
1199:             jsr      div32_16
1200:             ldd      quoLSB      * fin division
1201:             tst      buff
1202:             beq      s2_calcond
1203:             std      varph
1204:             ldd      condm
1205:             subd     varph
1206:             jmp      s3_calcond
1207: s2_calcond   addd     condm      * addition de condm
1208: s3_calcond   std      condc      * conductivité corrigée
1209:             puly
1210:             rts
1211:
1212: *****
1213: *      Sprog MCONDG1 (mesure conductivité gamme 1 de 1000 à 10000 µS)
1214: *      condm = ((( 11250 * rescond ) / 4096 ) - 125 ) * ( coeft / 100 )
1215: *****
1216: mcondg1      pshy
1217:             ldx      #11250
1218:             ldd      rescond
1219:             jsr      mul16_16    * 11250 * rescond
1220:             pshx
1221:             tsx
1222:             xgdy      * Resultatsur 24 bits
1223:             tab      * division par 4096
1224:             ldaa     1,x
1225:             ror      0,x
1226:             rora
1227:             rorb
1228:             ror      0,x
1229:             rora
1230:             rorb
1231:             ror      0,x
1232:             rora
1233:             rorb
1234:             ror      0,x
1235:             rora
1236:             rorb
1237:             pulx      * Fin division
1238:             subd     #125        * on retire 125
1239:             xgdx
1240:             ldd      coeft      * coeft: coefficient cellule de la sonde
1241:             jsr      mul16_16    * multiplication par coeft
1242:             stx      dividMSB    * début division par 100
1243:             sty      dividLSB
1244:             ldd      #100
1245:             std      divisLSB
1246:             ldd      #0
1247:             std      divisMSB
1248:             ldx      #quoMSB

```

```

1249:      jsr      div32_16
1250:      ldd      quoLSB          * fin division
1251:      std      condm
1252:      puly
1253:      rts
1254:
1255: *****
1256: *      Sprog MCONDG2      (mesure conductivité gamme 2 de 100 à 1000 µS)
1257: *      condm = ((( 1125 * rescond ) / 4096 ) - 12,5 ) * ( coefc / 100 )
1258: *****
1259: mcondg2      pshy
1260:              ldx      #4500          * ( 1125 * 4 )
1261:              ldd      rescond
1262:              jsr      mul16_16      * 1125 * rescond * 4
1263:              pshx
1264:              tsx
1265:              xgdy          * Resultatsur 24 bits
1266:              tab          * division par 4096
1267:              ldaa      1, x
1268:              ror      0, x
1269:              rora
1270:              rorb
1271:              ror      0, x
1272:              rora
1273:              rorb
1274:              ror      0, x
1275:              rora
1276:              rorb
1277:              ror      0, x
1278:              rora
1279:              rorb
1280:              pulx          * Fin division
1281:              subd      #50          * on retire 50 (12,5 * 4)
1282:              xgdx
1283:              ldd      coefc          * coefc: coefficient cellule de la sonde
1284:              jsr      mul16_16      * multiplication par coefc
1285:              stx      dividMSB      * début divisoin par 4 * 100
1286:              sty      dividLSB
1287:              ldd      #400
1288:              std      divisLSB
1289:              ldd      #0
1290:              std      divisMSB
1291:              ldx      #quoMSB
1292:              jsr      div32_16
1293:              ldd      quoLSB          * fin division
1294:              std      condm
1295:              puly
1296:              rts
1297:
1298:
1299: *****
1300: *  procédures utilisées pour le calcul pH, conductivité et température
1301: *****
1302:
1303: *      *****
1304: *      *      PROCEDURE MUL16_16      *
1305: *      *****
1306: *  Multiplication non signee 16 bits. Le registre D contient le multiplicande
1307: *  le regitre X contient le multiplicateur. En sortie le resultat est place
1308: *  dans les registre X(MSB) et Y(LSB).
1309: *
1310: *  Registres Affect,s: X, Y, CCR
1311: *  Points d'entrée : D et X.
1312: *  Points de sortie : X et Y (X:Y=D*X)
1313:
1314: mul16_16      pshb          * Sauvegarde de D
1315:              psha
1316:
1317:              ldy      #cal16x16
1318:              std      4, Y          * Mise en place du multiplicande a
1319:              ldd      #0          * decaler sur 32 bits.
1320:              std      2, Y
1321:              std      6, Y          * Initialisation du resultat a 0.
1322:              std      8, Y
1323:              xgdx          * X contenait le multiplicateur
1324: b0_mul16_16      lsrld
1325:              std      0, Y          * Sauvegarde du multiplicateur decale.
1326:              bcc      s0_mul16_16  * Doit additionner le resultat au

```

```

1327: *                                * multiplicande decale.
1328:         ldd      4,Y              * Oui alors addition sur 32 bits
1329:         addd     8,Y
1330:         std      8,Y
1331:         ldaa     3,Y
1332:         adca     7,Y
1333:         staa     7,Y
1334:         ldaa     2,Y
1335:         adca     6,Y
1336:         staa     6,Y
1337: s0_mul16_16  ldd      4,Y              * Decalage a gauche sur 32 bits du
1338:         lsld                                * multiplicande
1339:         std      4,Y
1340:         ldaa     3,Y
1341:         rola
1342:         staa     3,Y
1343:         ldaa     2,Y
1344:         rola
1345:         staa     2,Y
1346:         ldd      0,Y
1347:         bne      b0_mul16_16        * Doit arreter la multiplication car
1348: *                                * le multiplicateur = 0.
1349:         ldx      6,Y              * Mise en place du resultat dans X et
1350:         ldy      8,Y              * Y.
1351:         pula                                * Restitution de D
1352:         pulb
1353:         rts
1354:
1355:
1356: *                                *****
1357: *                                * PROCEDURE DIV32_16 *
1358: *                                *****
1359:
1360: * Division non signee 32*16 bits resultat sur 16 bits.
1361: * Registres Affect,s: D, X, Y, CCR
1362: * Points d'entree : X pointe sur une structure division
1363: * Structure division:
1364: *         quoMSB      rmb      2      * MSB quotient resultat. Modifié par div32_16
1365: *         quoLSB      rmb      2      * LSB quotient resultat
1366: *         reste_divM  rmb      2      * MSB reste resultat. Modifié par div32_16 (4)
1367: *         reste_divL  rmb      2      * LSB reste resultat
1368: *         divisMSB    rmb      2      * MSB numerateur. Non modifié par div32_16 (8)
1369: *         divisLSB    rmb      2      * LSB numerateur
1370: *         dividMSB    rmb      2      * MSB denominator. Modifié par div32_16 (12)
1371: *         dividLSB    rmb      2      * LSB denominator
1372: *
1373: * Points de sortie :
1374:
1375: div32_16      equ      *
1376:         ldd      #0
1377:         std      0,x
1378:         std      2,x
1379:         std      4,x
1380:         std      6,x
1381:         ldd      10,x
1382:         bne      s0_div32_16
1383:         ldd      #$FFFF              * Test Division par 0
1384:         std      0,x
1385:         std      2,x
1386:         rts
1387:
1388: s0_div32_16    pshy
1389:         ldy      #32
1390: b1_div32_16    lsl      15,x
1391:         rol      14,x
1392:         rol      13,x
1393:         rol      12,x
1394:         ldd      6,x
1395:         rolb
1396:         rola
1397:         subd     10,x
1398:         blo      s1_div32_16
1399:         std      6,x
1400:         sec
1401:         rol      3,x
1402:         rol      2,x
1403:         bra      s2_div32_16
1404: s1_div32_16    addd     10,x

```

```

1405:          std      6,x
1406:          lsl      3,x
1407:          rol      2,x
1408: s2_div32_16  dey
1409:          bne      b1_div32_16
1410:          puly
1411:          rts
1412:
1413:
1414:
1415: * Division non signee 32*32 bits resultat sur 32 bits.
1416: * Registres Affect,s: D, X, Y, CCR
1417: * Points d'entr,e : X pointe sur une structure division
1418: * Structure division:
1419: *      quoMSB      rmb      2      * Modifi, par div32
1420: *      quoLSB      rmb      2
1421: *      reste_divM  rmb      2      * Modifi, par div32 (4)
1422: *      reste_divL  rmb      2
1423: *      divisMSB    rmb      2      * Non modifi, par div32 (8)
1424: *      divisLSB    rmb      2
1425: *      dividMSB    rmb      2      * Modifi, par div32 (12)
1426: *      dividLSB    rmb      2
1427: *
1428: * Points de sortie :
1429: * Dur,e :
1430:
1431:
1432: div32_32     equ      *
1433:          ldd      #0
1434:          std      0,x
1435:          std      2,x
1436:          std      4,x
1437:          std      6,x
1438:          ldd      8,x      * Test division par 0
1439:          bne      s0_div
1440:          ldd      10,x
1441:          bne      s0_div
1442:          ldd      #FFFF    * Division par 0
1443:          std      0,x
1444:          std      2,x
1445:          rts
1446:
1447: s0_div       pshy
1448:          ldy      #32
1449: b1_div       lsl      15,x
1450:          rol      14,x
1451:          rol      13,x
1452:          rol      12,x
1453:          rol      7,x
1454:          rol      6,x
1455:          rol      5,x
1456:          rol      4,x
1457:          ldd      4,x
1458:          cpd      8,x
1459:          blo      s1_div
1460:          ldd      6,x
1461:          cpd      10,x
1462:          blo      s1_div
1463:          subd     10,x
1464:          std      6,x
1465:          ldaa     5,x
1466:          sbca     9,x
1467:          staa     5,x
1468:          ldaa     4,x
1469:          sbca     8,x
1470:          staa     4,x
1471:          sec
1472:          bra      s2_div
1473: s1_div       clc
1474: s2_div       rol      3,x
1475:          rol      2,x
1476:          rol      1,x
1477:          rol      0,x
1478:          dey
1479:          bne      b1_div
1480:          puly
1481:          rts
1482:

```

```

1483:
1484: *****
1485: ***      SProgs de gestion du CAN      *
1486: *****
1487:
1488: *****
1489: * SP      lancement conversion température
1490: * E: -    S: restemp (16bits)      modif: D
1491: *****
1492: lcant      ldd      #0
1493:            std      moyenne
1494:            ldx      #16
1495: s0_lcant   pshx
1496:            bclr     porta,y memcan ; selection CAN
1497:            ldab     #%10010111 ; com: B=
1498:            jsr      comcan
1499:            jsr      tp50us
1500:            jsr      lectr
1501:            ldd      result
1502:            bset     porta,y memcan ; deselection CAN
1503:            addd     moyenne
1504:            std      moyenne
1505:            pulx
1506:            dex
1507:            bne      s0_lcant
1508:            ldd      moyenne
1509:            lsr      lsr
1510:            lsr      lsr
1511:            lsr      lsr
1512:            lsr      lsr
1513:            std      restemp
1514:            rts
1515:
1516: *****
1517: * SP      lancement conversion pH
1518: * E: -    S: resph (16bits)      modif: D
1519: *****
1520: lcanp      ldd      #0
1521:            std      moyenne
1522:            ldx      #16
1523: s0_lcanp   pshx
1524:            bclr     porta,y memcan ; selection CAN
1525:            ldab     #%11010111 ; com: B=
1526:            jsr      comcan
1527:            jsr      tp50us
1528:            jsr      lectr
1529:            ldd      result
1530:            bset     porta,y memcan ; deselection CAN
1531:            addd     moyenne
1532:            std      moyenne
1533:            pulx
1534:            dex
1535:            bne      s0_lcanp
1536:            ldd      moyenne
1537:            lsr      lsr
1538:            lsr      lsr
1539:            lsr      lsr
1540:            lsr      lsr
1541:            std      resph
1542:            rts
1543:
1544: *****
1545: * SP      lancement conversion conductivité
1546: * E: -    S: rescond (16bits)      modif: D
1547: *****
1548: lcanc      ldd      #0
1549:            std      moyenne
1550:            ldx      #16
1551: s0_lcanc   pshx
1552:            bclr     porta,y memcan ; selection CAN
1553:            ldab     #%10100111 ; com: B=
1554:            jsr      comcan
1555:            jsr      tp50us
1556:            jsr      lectr
1557:            ldd      result
1558:            bset     porta,y memcan ; deselection CAN
1559:            addd     moyenne
1560:            std      moyenne

```

```

1561:                pulx
1562:                dex
1563:                bne      s0_lcanc
1564:                ldd      moyenne
1565:                lsrđ
1566:                lsrđ
1567:                lsrđ
1568:                lsrđ
1569:                std      rescond
1570:                rts
1571:
1572:
1573: *****
1574: * SP      eécriture commande CAN
1575: *      E: B = octet de controle , S:-
1576: *****
1577: comcan                bclr      porta,y memcan * verification selection can
1578:                bclr      porta,y hdata * hdata=0
1579:                ldaa      #8
1580:                staa      compte ; initialisation compteur
1581: comc0
1582:                lslb
1583:                bcs      comc1 ; C=1 => din_can=1
1584:                bclr      porta,y din_can ; C=0 => din_can=0
1585:                bra      comc2
1586: comc1                bset      porta,y din_can
1587: comc2                bset      porta,y hdata * hdata=1 impulsion
1588:                bclr      porta,y hdata * hdata=0 sur hdata
1589:                dec      compte
1590:                bne      comc0
1591:                bclr      porta,y din_can ; din_can=0
1592:                rts
1593:
1594: *****
1595: * SP      lecture donnee CAN
1596: *      E: - S: result (16bits)
1597: *****
1598: lectr                ldaa      #16
1599:                staa      compte ; initialisation compteur
1600:                clra
1601:                clrb
1602: lectd0                bset      porta,y hdata ; impulsion sur hdata
1603:                bclr      porta,y hdata ;
1604:                brclr      porta,y dout lectd1 ; test dout can
1605:                sec
1606:                bra      lectd2
1607: lectd1                clc
1608: lectd2                rolb ; recuperation bit Dout sur reg.D
1609:                rola
1610:                dec      compte
1611:                bne      lectd0
1612:                lsrđ
1613:                lsrđ
1614:                lsrđ
1615:                std      result ; recuperation donnee can
1616:                rts
1617:
1618:
1619: *****
1620: ***      SProgs de gestion de la memoire ***
1621: *****
1622:
1623: *****
1624: * SP      gestion de lecture des parametres dans la memoire eeprom
1625: *      E: - S: pente,ph0,tempe,fctemp,coeft,coefc,vseuil,znseuil,sseuil modif: D,X *
1626: *****
1627: lectmem clr      adeprom ; adresse eeprom = 0
1628:                ldaa      #9
1629:                staa      ctpara ; 9 parametres à lire en memoire
1630:                ldx      #pente ; chargement adresse du 1er parametre
1631: s0_lectmem        jsr      read ; lecture des parametres en memoire
1632:                ldd      donnee
1633:                std      0,x
1634:                inx
1635:                inx
1636:                inc      adeprom
1637:                dec      ctpara
1638:                bne      s0_lectmem

```

```

1639:                rts
1640:
1641: *****
1642: * SP gestion de l'écriture des paramètres de fonctionnement dans la memoire *
1643: * E: pente,ph0,tempe,fcftemp,coeft,coefc,vseuil,znseuil,sseuil S:- modif: D,X *
1644: *****
1645: ecritmem        ldaa    #9
1646:                staa    ctpara
1647:                clr     adeprom
1648:                ldz     #pente
1649:                jsr     wen                ; autorisation ecriture en memoire
1650: s0_ecritmem     ldd     0,x
1651:                std     donnee
1652:                jsr     write            ; ecriture parametres en memoire
1653:                inc     adeprom
1654:                inx
1655:                inx
1656:                dec     ctpara
1657:                bne     s0_ecritmem
1658:                jsr     wds                ; interdiction ecriture en memoire
1659:                rts
1660:
1661: *****
1662: * SP write enable eeprom
1663: *****
1664: wen             bset     porta,y memcan ; CS=1 selection eeprom
1665:                ldaa     #%00000100 ; com: A=xxxxx+1+00 B=xx+11+xxxx
1666:                ldab     #%00110000
1667:                jsr     writcom
1668:                bclr     porta,y memcan ; CS=0 deselection eeprom
1669:                rts
1670:
1671: *****
1672: * SP write disable eeprom
1673: *****
1674: wds             bset     porta,y memcan ; selection eeprom
1675:                ldaa     #%00000100 ; com: A=xxxxx+1+00 B=xx+00+xxxx
1676:                ldab     #%00000000
1677:                jsr     writcom
1678:                bclr     porta,y memcan ; deselection eeprom
1679:                rts
1680:
1681: *****
1682: * SP write eeprom
1683: * E: adeprom(8bits), donnee(16bits) S:
1684: *****
1685: write          bset     porta,y memcan ; selection eeprom
1686:                ldaa     #%00000101 ; com: A=xxxxx+1+01
1687:                ldab     adeprom
1688:                jsr     writcom
1689:                jsr     writdon
1690:                bclr     porta,y memcan ; deselection eeprom
1691:                jsr     finwrit ; attente fin d'écriture (Tbusy)
1692:                rts
1693:
1694: *****
1695: * SP ecriture commande eeprom (bit start+code fonction+adresse)
1696: * E: A=xxxxx+bit start+code fonction, B=xx+adresse(6bits)
1697: *****
1698: writcom        bset     porta,y memcan * verification selection eeprom
1699:                bclr     porta,y hdata * hdata=0
1700:                psha
1701:                ldaa     #9
1702:                staa     compte ; initialisation compteur
1703:                pula
1704:                lslb
1705:                lslb
1706:                lsld                * suppression 5 bits de poids fort
1707:                lsld
1708:                lsld
1709:                lsld
1710:                lsld
1711: writc0         lsld
1712:                bcs     writc1 ; C=1 => din_mem=1
1713:                bclr     porta,y din_mem ; C=0 => din_mem=0
1714:                bra     writc2
1715: writc1         bset     porta,y din_mem
1716: writc2         bset     porta,y hdata * hdata=1 impulsion

```

```

1717:          bclr    porta,y hdata    * hdata=0    sur hdata
1718:          dec     compte
1719:          bne     writc0
1720:          rts
1721:
1722: *****
1723: * SP      ecriture donnee eeprom
1724: * E: donnee(16bits)
1725: *****
1726: writdon          ldaa    #16
1727:                  staa    compte          ; initialisation compteur
1728:                  ldd     donnee
1729: writd0          lsld
1730:                  bcs     writd1          ; C=1 => din_mem=1
1731:                  bclr    porta,y din_mem ; C=0 => din_mem=0
1732:                  bra     writd2
1733: writd1          bset     porta,y din_mem
1734: writd2          bset     porta,y hdata    * hdata=1    impulsion
1735:                  bclr    porta,y hdata    * hdata=0    sur hdata
1736:                  dec     compte
1737:                  bne     writd0
1738:                  bclr    porta,y din_mem ; din_mem=0
1739:                  rts
1740:
1741: *****
1742: * SP      Attente fin d'une fonction d'écriture/effacement (busy/ready)
1743: * (Attente maximum = 4096.(3+8+3).Tcycle )
1744: *****
1745: finwrit          pshx
1746:                  ld      #$1000
1747:                  bclr    porta,y memcan
1748:                  bset     porta,y memcan
1749: finwri0          dex
1750:                  brset    porta,y dout finwri1 ; attente de l'etat ready
1751:                  bne     finwri0
1752: finwri1          bclr    porta,y memcan ; deselection du CS apres le ready
1753:                  pulx
1754:                  rts
1755:
1756: *****
1757: * SP      read eeprom
1758: * E: adeprom(8bits) S: donnee (16bits)
1759: *****
1760: read             bset     porta,y memcan ; selection eeprom
1761:                  ldaa    #%00000110 ; com: A=xxxxx+1+10
1762:                  ldab    adeprom
1763:                  jsr     writcom
1764:                  jsr     readdon
1765:                  std     donnee ; recuperation donnee eeprom
1766:                  bclr    porta,y memcan ; deselection eeprom
1767:                  rts
1768:
1769: *****
1770:
1771: * SP      lecture donnee eeprom
1772: * E: - S: donnee (16bits)
1773: *****
1774: readdon          ldaa    #16
1775:                  staa    compte          ; initialisation compteur
1776: readd0          bset     porta,y hdata    ; impulsion sur hdata
1777:                  bclr    porta,y hdata    ;
1778:                  brclr    porta,y dout readd1 ; test dout eeprom
1779:                  sec
1780:                  bra     readd2
1781: readd1          clc
1782: readd2          rolb          ; recuperation bit Dout sur reg.D
1783:                  rola
1784:                  dec     compte
1785:                  bne     readd0
1786:                  std     donnee ; recuperation donnee eeprom
1787:                  rts
1788:
1789: *****
1790:
1791: * SProg lecture du clavier
1792: * E: - S: seuil, phcond,etaltemp,etalph,etalcond,moins,plus,val
1793: *****
1794: lec_clavier      brset    porte,y t_clav f_lec_clav ; test appui sur un BP sinon fin du

```

```

1795:          ldaa     porte                ; lecture du code du BP
1796:          anda     #%00000111
1797:          staa     menu                ; menu = valeur porteE
1798: s0_lec_clavier  ldaa     menu
1799:          cmpa     #%00000000
1800:          bne      s1_lec_clavier      ; saut à étiquette si pas d'appui sur touche
1801:          ldaa     #$FF                ; touche seuil appuyee
1802:          staa     seuil                ; si menu = 0 alors seuil = $FF
1803: s1_lec_clavier  ldaa     menu
1804:          cmpa     #%00000001
1805:          bne      s2_lec_clavier      ; saut à étiquette si pas d'appui sur touche
1806:          tst      phcond                ; touche phcond appuyee
1807:          beq      menuphc              ; si menu = 1 alors complement de phcond
1808:          clr      phcond
1809:          jsr      clr_aff
1810:          jmp      s2_lec_clavier
1811: menuphc  ldaa     #$FF
1812:          staa     phcond
1813:          jsr      clr_aff
1814: s2_lec_clavier  ldaa     menu
1815:          cmpa     #%00000010
1816:          bne      s3_lec_clavier      ; saut à étiquette si pas d'appui sur touche
1817:          ldaa     #$FF                ; touche etaltemp appuyee
1818:          staa     etaltemp            ; si menu = 2 alors etaltemp = $FF
1819: s3_lec_clavier  ldaa     menu
1820:          cmpa     #%00000011
1821:          bne      s4_lec_clavier      ; saut à étiquette si pas d'appui sur touche
1822:          ldaa     #$FF                ; touche etalph appuyee
1823:          staa     etalph              ; si menu = 3 alors etalph = $FF
1824: s4_lec_clavier  ldaa     menu
1825:          cmpa     #%00000100
1826:          bne      s5_lec_clavier      ; saut à étiquette si pas d'appui sur touche
1827:          ldaa     #$FF                ; touche etalcond appuyee
1828:          staa     etalcond            ; si menu = 4 alors etalcond = $FF
1829: s5_lec_clavier  ldaa     menu
1830:          cmpa     #%00000101
1831:          bne      s6_lec_clavier      ; saut à étiquette si pas d'appui sur touche
1832:          ldaa     #$FF                ; touche moins appuyee
1833:          staa     moins                ; si menu = 5 alors moins = $FF
1834: s6_lec_clavier  ldaa     menu
1835:          cmpa     #%00000110
1836:          bne      s7_lec_clavier      ; saut à étiquette si pas d'appui sur touche
1837:          ldaa     #$FF                ; touche plus appuyee
1838:          staa     plus                ; si menu = 6 alors plus = $FF
1839: s7_lec_clavier  ldaa     menu
1840:          cmpa     #%00000111
1841:          bne      fin_lec_clav        ; saut à étiquette si pas d'appui sur touche
1842:          ldaa     #$FF                ; touche val appuyee
1843:          staa     val                  ; si menu = 7 alors plus = $FF
1844: fin_lec_clav   brclr  porte,y t_clav fin_lec_clav ; test relâchement bouton poussoir
1845: f_lec_clav     rts
1846:
1847: *****
1848: *          Sprog gestion menu
1849: *  E: seuil,etaltemp,etalph,etalcond      S:-
1850: *****
1851: ges_menu      tst      etaltemp        ; test appui touche etaltemp
1852:          beq      s0_ges_menu          ; saut à étiquette si pas d'appui sur touche
1853:          clr      etaltemp              ; etalonnage sonde temperature si
1854:          jsr      etalon_temp            ; touche etaltemp appuyee
1855:          jsr      ecrimem                ; ecriture parametres de fonctionnement en m
1856:          jmp      fin_ges_menu
1857: s0_ges_menu   tst      etalph          ; test appui touche etalph
1858:          beq      s1_ges_menu          ; saut à étiquette si pas d'appui sur touche
1859:          clr      etalph                ; etalonnage sonde pH si
1860:          jsr      etalon_ph              ; touche etalph appuyee
1861:          jsr      ecrimem                ; ecriture parametres de fonctionnement en m
1862:          jmp      fin_ges_menu
1863: s1_ges_menu   tst      etalcond        ; test appui touche etalcond
1864:          beq      s2_ges_menu          ; saut à étiquette si pas d'appui sur touche
1865:          clr      etalcond              ; etalonnage sonde conductivite si
1866:          jsr      etalon_cond            ; touche etalcond appuyee
1867:          jsr      ecrimem                ; ecriture parametres de fonctionnement en m
1868:          jmp      fin_ges_menu
1869: s2_ges_menu   tst      seuil            ; test appui touche seuil
1870:          beq      fin_ges_menu          ; saut à étiquette si pas d'appui sur touche
1871:          clr      seuil                  ; reglages seuil pH si
1872:          jsr      ges_seuil              ; touche seuil appuyee

```

```

1873:                jsr      ecritmem                ; ecriture parametres de fonctionnement en m
1874: fin_ges_menu    rts
1875:
1876:
1877: *****
1878: *      gestion seuil
1879: *****
1880: ges_seuil        nop                        ; à définir
1881:                rts
1882:
1883:
1884: *****
1885: ***      SProg evoie ph boucle de courant
1886: * e: restemp    s: valth, valtl modif: d, tmsk1, x, y
1887: *****
1888: envph            pshy
1889:                ldd      phcor
1890:                ldx      #2761
1891:                jsr      mul16_16
1892:                stx      dividMSB
1893:                sty      dividLSB
1894:                ldd      #10
1895:                std      divisLSB
1896:                ldd      #0
1897:                std      divisMSB
1898:                ldx      #quoMSB
1899:                jsr      div32_16
1900:                ldd      quoLSB
1901:                addd     #9659
1902:                std      valth
1903:                ldd      #60387
1904:                subd     valth
1905:                std      valtl
1906:                ldaa     #%100000000
1907:                staa     tmsk1
1908:                cli
1909:                puly
1910:                rts
1911:
1912:
1913: *****
1914: ***      Sprog d'interruption timer 1
1915: *** E: valtl, valth    S:      modif: btimer, d, tmsk1, toc1
1916: *****
1917: spinter1        pshy
1918:                ldy      #$1000
1919:                tst      btimer
1920:                bne      inter1
1921:                ldd      valth
1922:                bset     portc, y paramrap
1923:                inc      btimer
1924:                bra      fininter
1925: inter1          ldd      valtl
1926:                bclr     portc, y paramrap
1927:                clr      btimer
1928: fininter        addd     toc1
1929:                std      toc1
1930:                ldaa     #%100000000
1931:                staa     tflg1
1932:                puly
1933:                rti
1934:
1935:
1936: *****
1937: ***      SProgs de temporisation
1938: *****
1939:
1940: *****
1941: *      temporisation de 50us
1942: *****
1943: tp50us          pshx
1944:                ldx      #11
1945: tp50us1         dex
1946:                bne      tp50us1
1947:                pulx
1948:                rts
1949:
1950: *****

```

```

1951: *          temporisation de 1ms
1952: *****
1953: tp1ms          pshx
1954:                ldx          #304
1955: tp1ms1         dex
1956:                bne          tp1ms1
1957:                pulx
1958:                rts
1959:
1960:
1961: *****
1962: *          temporisation de 10ms
1963: *****
1964: tp10ms          pshx
1965:                ldx          #3069
1966: tp10ms1         dex
1967:                bne          tp10ms1
1968:                pulx
1969:                rts
1970:
1971: *****
1972: *          temporisation de 100ms
1973: *****
1974: tp100ms         pshx
1975:                ldx          #30718
1976: tp100m1         dex
1977:                bne          tp100m1
1978:                pulx
1979:                rts
1980:
1981: *****
1982:
1983: * procedure de temporisation de 1 s
1984: *****
1985: tp1s            pshx
1986:                ldx          #10
1987: tp1s1           jsr          tp100ms
1988:                dex
1989:                bne          tp1s1
1990:                pulx
1991:                rts
1992:
1993:                END          *** Fin du prog en RAM ***

```