```
/*****************************************************************/
/*!
 * Edition du message par défaut
 * Si E4 est pressé, sort plus tôt pour éviter de sauvegarder le message
 * E1 incrémente le pointeur de caractère
 * E2 décrémente le pointeur de caractère
 * E3 bascule du blocage de majuscule
 * E4 sort sans sauvegarder le message
 * E5 enregistre le caractère et va au caractère suivant
 * E6 efface le caractère précédent
 * E8 sauvegarde le message
 */

void
EditMessage()
{
  int i;
  char scrollspeed; // vitesse de défilement

  char message[MAX_CHAR_LENGTH]; // longueur de message maxi
  unsigned char alphabetpos=0; // position dans l'alphabet
  unsigned char messagepos=0; // position dans le message
  unsigned char CapsLock=0; // blocage de la touche majuscule

  for(i=0;i<MAX_CHAR_LENGTH;i++)
  {
    message[i]=0; // RAZ du tableau des caractères du message
  }
  message[0]='a'; // 'a' est affiché par une sous routine d'interruption tant qu'on appuie p

  /* Attend jusqu'à ce que une touche quelconque ait été pressée */
  /* KeyPressed renvoie 1 si une touche est pressée, la boucle ne fait rien)*/
  while(KeyPressed()) {
  }

  /* Boucle jusqu'à ce que la touche 8 soit pressée */
  while(LastKeyPressed()!=8)
  {
    if(KeyPressed())
    {
      /* incrémentation du pointeur de caractère */
      if(LastKeyPressed()==1) // par appui sur la première touche
      {
        alphabetpos++; // incrémentation du pointeur de position dans l'alphabet
        /* Bouclage de la position du pointeur s'il dépasse la dimension du tableau */
        if(alphabetpos>=sizeof(SelectChar))
        {
          alphabetpos=0;
        }

        /* Si le blocage des majuscules est actif, on utilise les lettres capitales */
        if(CapsLock==1 && SelectChar[alphabetpos]>='a' && SelectChar[alphabetpos]<='z')
        {
           message[messagepos]=SelectChar[alphabetpos]-32; // décalage du pointeur de caract
        }
        else
        {
          message[messagepos]=SelectChar[alphabetpos]; // sinon on compose le message avec l
        }
      }

      /* Décrémentation du pointeur de caractère */
      if(LastKeyPressed()==2) // par l'appui sur la touche 2
      {
        /* Bouclage de la position du pointeur s'il dépasse la dimension du tableau */
        if(alphabetpos==0)
        {
          alphabetpos=sizeof(SelectChar)-1;
        }
        else
        {
          alphabetpos--; // décrémentation du pointeur
        }
        /* Si le blocage des majuscules est actif, on utilise les lettres capitales */
        if(CapsLock==1 && SelectChar[alphabetpos]>='a' && SelectChar[alphabetpos]<='z')
        {
```

```
77              message[messagepos]=SelectChar[alphabetpos]-32; // décalage du pointeur de caract
78            }
79            else
80            {
81              message[messagepos]=SelectChar[alphabetpos]; // sinon on compose le message avec l
82            }
83          }
84
85          /* Bascule du blocage des majuscules */
86          if(LastKeyPressed()==3) // par appui sur la touche 3
87          {
88            if(CapsLock==1) // si on était en mode majuscule
89            {
90              message[messagepos]=SelectChar[alphabetpos]; // on ne modifie pas le pointeur d'éc
91              CapsLock=0; // et on met le mode en minuscule
92            }
93            else
94            {
95              if(SelectChar[alphabetpos]>='a' && SelectChar[alphabetpos]<='z') // si le pointeur
96              {
97                message[messagepos]=SelectChar[alphabetpos]-32; // On effectue un déplacement du
98              }
99              CapsLock=1; // et on met le mode en majuscule
100
101           }
102         }
103
104         /* Suppression d'un caractère */
105         if(LastKeyPressed()==4) // par appui sur la touche 4
106         {
107           return;
108         }
109
110         /* Enregistrement du caractère */
111         if(LastKeyPressed()==5) // par appui sur la touche 5
112         {
113           if(messagepos < (MAX_CHAR_LENGTH - 1)) // à condition qu'on ne dépasse pas la longue
114           {
115
116             messagepos++;
117
118             if(CapsLock==1 && SelectChar[alphabetpos]>='a' && SelectChar[alphabetpos]<='z')
119             {
120               message[messagepos]=SelectChar[alphabetpos]-32; // écriture en majuscule
121             }
122             else
123             {
124               message[messagepos]=SelectChar[alphabetpos]; // écriture en minuscule
125             }
126           }
127
128         }
129
130         /* Suppression du dernier caractère */
131         if(LastKeyPressed()==6) // par appui sur la touche 6
132         {
133           if(messagepos>0)
134           {
135             message[messagepos]=0;
136             messagepos--;
137             //message[messagepos]=0;
138           }
139         }
140
141         /* Quand une touche quelconque est appuyée et qu'on ne fait rien d'autre on affiche le
142           /* petit rafraichissement visuel pendant l'appui */
143         while(KeyPressed())
144         {
145           LEDWrite(message);
146         }
147       }
148       else
149       {
150         LEDWrite(message); // si on n'est pas dans une phase de composition du message on affi
151       }
152     }
```

```
153
154    /* Frappe de la touche de sortie. Sauvegarde du dernier caractère */
155    messagepos++;
156    message[messagepos]=0; // 0 en dernière position
157
158
159    scrollspeed=*SpeedAddress;
160
161    /* Ecriture du message en mémoire Flash */
162    if(message[0]!=0 && message[0]!=' ') // s'il y a quelque chose à sauvegarder
163    {
164      SCGC2_FLS=1;
165      Flash_SectorErase(MessageAddress);
166      Flash_ByteProgram(MessageAddress,message,(MAX_CHAR_LENGTH/4 + 1));
167      Flash_ByteProgram(SpeedAddress,&scrollspeed,1);
168      SCGC2_FLS=0;
169    }
170  }
171
```