

```

1  program TEST_RS232;
2
3  {$NOSHADOW}
4  { $WG}                                {global Warnings off}
5
6  Device = mega88, VCC=5;
7  { $BOOTRST $00C00}                    {Reset Jump to $00C00}
8
9  Define_Fuses
10 //  Override_Fuses;
11     Notebook    = A;
12     COMport     = USB;
13     LockBits0   = [];
14     FuseBits0   = [CKSEL0, CKSEL2, CKSEL3]; // Horloge interne 8 MHz
15     FuseBits1   = [];
16     FuseBits2   = [];
17
18 Import SysTick, SerPort, LCDport, ADCPort;
19
20
21
22 From System Import Float;
23
24
25 Define
26     ProcClock    = 8000000;             {Hertz}
27     SysTick      = 10;                   {msec}
28     StackSize    = $0032, iData;
29     FrameSize    = $0032, iData;
30     SerPort      = 9600, Stop1;         {Baud, StopBits|Parity}
31     RxBuffer     = 100, iData;
32     TxBuffer     = 100, iData;
33     LCDport      = PortB,0,PortB,4;
34     LCDtype      = 44780;
35     LCDrows      = 4;                   {rows}
36     LCDcolumns   = 20;                   {columns per line}
37     ADCchans     = [6], iData;
38     ADCpresc     = 128;
39
40 Implementation
41
42 {$IDATA}
43
44 {-----}
45 { Type Declarations }
46
47 type
48
49
50 {-----}
51 { Const Declarations }
52
53 {-----}
54 { Var Declarations }
55 {$IDATA}
56 var
57

```

```

58 // Entrées boutons poussoirs
59 BP1[@PinC, 2] : bit; // Bouton poussoir
60 BP2[@PinC, 1] : bit; // Bouton poussoir
61 BP3[@PinC, 0] : bit; // Bouton poussoir
62
63 // Gestion des boutons poussoirs, bit de mémorisation, bit d'appui
64 BP : byte;
65 BP1_mem[@BP, 0] : bit;
66 BP1_appui[@BP, 1] : bit;
67 BP2_mem[@BP, 2] : bit;
68 BP2_appui[@BP, 3] : bit;
69 BP3_mem[@BP, 4] : bit;
70 BP3_appui[@BP, 5] : bit;
71
72 // Pour le convertisseur ADC
73 ADC_Channel : word;
74 ADC_Low[@ADC_Channel] : byte;
75 ADC_Hi[@ADC_Channel+1] : byte;
76 ADC_Channel_Integer : integer absolute ADC_Channel;
77
78
79 // Variables diverses
80 i : byte;
81 Receive : byte;
82 Count : byte;
83
84 {-----}
85 { functions }
86
87 procedure InitPorts;
88 begin
89     PortC:= %00000111;
90 end InitPorts;
91
92
93 procedure TestBP;
94 begin
95     if ( not(BIT(BP1)) and BIT(BP1_mem) )
96     then BP1_appui:=true;
97     else BP1_appui:=false;
98     endif;
99     BP1_mem := BP1;
100
101     if ( not(BIT(BP2)) and BIT(BP2_mem) )
102     then BP2_appui:=true;
103     else BP2_appui:=false;
104     endif;
105     BP2_mem := BP2;
106
107     if ( not(BIT(BP3)) and BIT(BP3_mem) )
108     then BP3_appui:=true;
109     else BP3_appui:=false;
110     endif;
111     BP3_mem := BP3;
112 end;
113
114 {-----}

```

```

115 { Main Program }
116 {$IDATA}
117
118 begin
119     InitPorts;
120     EnableInts;
121     Count := 0;
122
123     LcdXY(0,0);
124     Write(LcdOut, '          AVRco          ');
125     LcdXY(0,1);
126     Write(LcdOut, '    ELAB Computer    ');
127     mDelay(200);
128
129     for i:=0 to 19 do
130         LcdXY(i,3);Write(LcdOut, '* ');
131         mDelay(50);
132     endfor;
133     LcdClr;
134     Write(LcdOut, 'ESSAI LIAISON SERIE');
135     LcdXY(0,1);
136     Write(LcdOut, 'P.G Vox          V1.0');
137     mDelay(2000);
138     LcdClr;
139
140     LcdXY(0,2);
141     Write(LcdOut, 'Caracteres recus');
142
143     loop
144         TestBP;
145
146
147         ADC_Channel := GetAdc and $3FF;
148         LcdXY(0,0); Write(LCDout, 'VCAN: '+IntToStr(ADC_Channel_Integer:4)+
149             ' =>'+FloatToStr((Float(ADC_Channel)*4.8828125e-03):6:4)+'V');
150
151
152         if (BIT(BP1_appui)) then
153             LcdXY(0,3);
154             Write(LcdOut, ' ');
155             Count:=0;
156         endif;
157
158         if (BIT(BP2_appui)) then
159             LcdXY(0,1);
160             Write(LcdOut, 'Envoi Bonjour ');
161             SerOUT('B');SerOUT('o');SerOUT('n');SerOUT('j');
162             SerOUT('o');SerOUT('u');SerOUT('r');
163             mDelay(2000);
164             LcdXY(0,1);
165             Write(LcdOut, ' ');
166         endif;
167
168
169         if (BIT(BP3_appui)) then
170             LcdXY(0,1);
171             Write(LcdOut, 'Envoi CAN');SerOUT(ADC_Hi);SerOUT(ADC_Low);

```

```
172         mDelay(2000);
173         LcdXY(0,1);
174         Write(LcdOut, ' ');
175     endif;
176
177
178     if SerStat then
179         Receive:=Serinp;
180         LcdXY(Count,3);
181         Write(LcdOut,Receive);
182         Count:=Count+1;
183         if (Count=20) then
184             mDelay(1000);
185             LcdXY(0,3);
186             Write(LcdOut, ' ');
187             Count:=0;
188         endif;
189
190     endif;
191
192 endloop;
193 end TEST_RS232.
194
195
```