



Réveil



Projet d'étude et réalisation

Plan

- Réveil classique
- Cahier des charges
- Notre projet
- Conclusion
- Bibliographie



Réveil disponible dans le commerce



Différentes fonctions usuelles

- Affichage de l'heure
- Réglage de l'heure
- Alarme
- Réglage de l'alarme
- Radio
- Affichage de la date
- Adaptation de luminosité

...



Les nouvelles générations de réveils

- Ajouts de nouvelles fonctions



Cahiers des charges

- Fonctions retenues
- Fournitures prévues
- Planning prévisionnel



Planning prévisionnel

S03/04/05/06	1ère fonction, étude du montage, implantation du programme
S07/08/09/10	Faire un typhon avec les boutons, le snoozer ainsi que la photorésistance
S11/12	Finalisation et derniers réglages, améliorations de la syntaxe du programme
S13/14	Rédaction du rapport, clôture du projet, améliorations du programme

Projet : Réalisation d'un réveil

- Objectif premier:
 - Obtenir une clock
 - Recherche de solutions

Utilisation d'un timer



Réalisation d'un timer



CodeWizardAVR - untitled.cwp

File Help

USART | Analog Comparator | ADC | SPI
I2C | 1 Wire | 2 Wire (I2C)
LCD | Bit-Banged | Project Information
Chip | Ports | External IRQ | Timers

Timer 0 | Timer 1 | Timer 2 | Watchdog

Clock Source: System Clock
Clock Value: 62,500 kHz
Mode: CTC top=OCR1A
Out. A: Discon. Out. B: Discon.

Input Capt. : Noise Cancel
Interrupt on: Compare A Match
Value: f4 h Inp. Capture: 0 h
Comp. A: 24 h B: 0 h

Programmation du timer

```
36 int heu;
37 int min;
38 int sec;
39 int alarme_H;
40 int alarme_Min;
41 int autorisation_reglage_alarme;
42 int autorisation_reglage_heure;
43
44 //déclaration des fonctions
45 void affichage (int heure, int minute, int seconde);
46 void reglage_H (void);
47 void reglage_A (void);
48
49
50
51 // timer 1 output compare A interrupt service routine
52 interrupt [TIM1_COMPA] void timer1_compa_isr(void)
53 {
54     sec=sec+1;
55
56     if(sec==60)
57     {
58         sec=0;
59         min=min+1;
60         if(min ==60)
61         {
62             min=0;
63             heu=heu+1;
64
65             if(heu==24)
66             {
67                 heu=0;
68                 \
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Programmation de l'heure

```
51 // Timer 1 output compare & interrupt service routine
52 interrupt [TIM1_COMPA] void timer1_compa_isr(void)
53 {
54     sec=sec+1;
55
56     if(sec==60)
57     {
58         sec=0;
59         min=min+1;
60         if(min ==60)
61         {
62             min=0;
63             heu=heu+1;
64
65             if(heu==24)
66             {
67                 heu=0;
68             }
69
70         }
71
72     }
73     if(a==0)
74     affichage (heu,min,sec);
75
76
77 }
```





```
66         {
67         heu=0;
68         }
69
70     }
71
72     }
73     if(a==0)
74     affichage (heu,min,sec);
75
76
77 }
78 void affichage (int heure, int minute, int seconde)
79 {
80
81     sprintf(time,"%2d:%2d:%2d",heure,minute,seconde);
82     lcd_gotoxy(0,1);
83     lcd_puts(time);
84
85 }
86
87
88
89 void reglage_H (void)
90 {
91     int detect_nivhautBP1, detect_nivhautBP2;
92
93     autorisation_reglage_alarme=0;
94     if(autorisation_reglage_heure==1)
95     {
96
97         if((PIND.1)==1)
98             detect_nivhaut_BP1=0;
99
100        if(((PIND.1)==0)&&(d==0))
101            {
102                detect_nivhautBP1=1;
103                if (heu==23)
```



```

245
246 while (1)
247     {
248
249         while (PIND.6
250             regl
251
252             while (PIND.7
253                 regl
254
255             );
256     }
88
89 void reglage_H (void)
90 {
91     int detect_nivhautBP1, detect_nivhautBP2;
92
93     autorisation_reglage_alarme=0;
94     if (autorisation_reglage_heure==1)
95     {
96
97         if ((PIND.1)==1)
98             detect_nivhaut_BP1=0;
99
100        if (((PIND.1)==0) && (detect_nivhaut_BP1==0))
101            {
102                detect_nivhautBP1=1;
103                if (heu==23)
104                    heu=0;
105                else heu =heu+1;
106            }
107
108
109
110        if ((PIND.2)==1)
111            detect_nivhautBP2=0;
112
113        if (((PIND.2)==0) && (detect_nivhautBP2==0))
114            {
115                detect_nivhautBP2=1;
116                if (min==59)
117                    min=0;
118                else min=min+1;
119            }
120
121
122
123        if ((detect_nivhautBP1==1) || (detect_nivhautBP2==1))
124            affichage (heu,min,sec);
125
126        autorisation_reglage_alarme=1;
127    }
128 }

```

IS

s : heu

s : min

ants





```
131 void reglage_A(void) |
132 {
133
134 affichage(alarme_H, alarme_Min, sec);
135
136 autorisation_reglage_heure=0;
137 if(autorisation_reglage_alarme==1)
138 {
139
140
141     if(PIND.3==0)
142     {
143         if(alarme_H==23)
144             alarme_H=0;
145         else alarme_H = alarme_H +1;
146     }
147     if(PIND.4==0)
148     {
149         if(alarme_Min==59)
150             alarme_Min=0;
151         else alarme_Min = alarme_Min+1;
152     }
153
154     affichage(alarme_H, alarme_Min, sec);
155
156 autorisation_reglage_heure=1;
157 }
158 }
159
```



```
66         {
67         heu=0;
68         }
69
70     }
71
72     }
73     if(a==0)
74     affichage(heu,min,sec);
75
76
77 }
78 void affichage (int heure, int minute, int seconde)
79 {
80
81     sprintf(time,"%2d:%2d:%2d",heure,minute,seconde);
82     lcd_gotoxy(0,1);
83     lcd_puts(time);
84
85 }
86
87
88
89 void reglage_H (void)
90 {
91     int detect_nivhautBP1, detect_nivhautBP2;
92
93     autorisation_reglage_alarme=0;
94     if(autorisation_reglage_heure==1)
95     {
96
97         if((PIND.1)==1)
98             detect_nivhaut_BP1=0;
99
100        if(((PIND.1)==0)&&(d==0))
101        {
102            detect_nivhautBP1=1;
103            if(heu==23)
```



```

164 void main(void)
165 {
166 //configuration des ports et initialisation de ceux-ci.
167 PORTA=0x00;
168 DDRA=0x00;
169
170 PORTB=0x00;
171 DDRB=0x00;
172
173 PORTC=0x00;
174 DDRC=0x00;
175
176 PORTD=0x00;
177 DDRD=0x00;
178
179 // Timer/Counter 0 initialization
180 // Clock source: System Clock
181 // Clock value: Timer 0 Stopped
182 // Mode: Normal top=FFh
183 // OCO output: Disconnected
184 TCCR0=0x00;
185 TCNT0=0x00;
186 OCRO=0x00;
187
188 // Timer/Counter 1 initialization
189 // Clock source: System Clock
190 // Clock value: 62,500 kHz
191 // Mode: CTC top=OCR1A
192 // OC1A output: Discon.
193 // OC1B output: Discon.
194 // Noise Canceler: Off
195 // Input Capture on Falling Edge
196 // Timer 1 Overflow Interrupt: Off
197 // Input Capture Interrupt: Off
198 // Compare A Match Interrupt: On
199 // Compare B Match Interrupt: Off
200 TCCR1A=0x00;
201 TCCR1B=0x0C;
202 TCNT1H=0x00;
203 TCNT1L=0x00;
204 ICR1H=0x00;
205 ICR1L=0x00;
206 OCR1AH=0xF4;
207 OCR1AL=0x24;
208 OCR1BH=0x00;

```

```

209 OCR1BL=0x00;
210
211 // Timer/Counter 2 initialization
212 // Clock source: System Clock
213 // Clock value: Timer 2 Stopped
214 // Mode: Normal top=FFh
215 // OC2 output: Disconnected
216 ASSR=0x00;
217 TCCR2=0x00;
218 TCNT2=0x00;
219 OCR2=0x00;
220
221 // External Interrupt(s) initialization
222 // INT0: Off
223 // INT1: Off
224 // INT2: Off
225 MCUCR=0x00;
226 MCUCSR=0x00;
227
228 // Timer(s)/Counter(s) Interrupt(s) initialization
229 TIMSK=0x10;
230
231 // Analog Comparator initialization
232 // Analog Comparator: Off
233 // Analog Comparator Input Capture by Timer/Counter 1: Off
234 ACSR=0x80;
235 SFIOR=0x00;
236
237 // initialisation de l'écran LCD
238 lcd_init(15);
239 lcd_clear();
240 // Global enable interrupts
241 #asm("sei")
242
243 while (1)
244 {
245
246     while (PIND.6==0)
247         reglage_H();
248
249     while (PIND.7==0)
250         reglage_A();
251
252 };
253 }

```


Déclenchement de l'alarme

- Comparaison entre les variables de l'heure et de l'alarme
- Déclenchement sonore



Résultat de notre travail

- Fonctions principales du projet réalisées
- Quelques problèmes persistent malgré tout
- Finalisation du prototype réalisable dans les temps

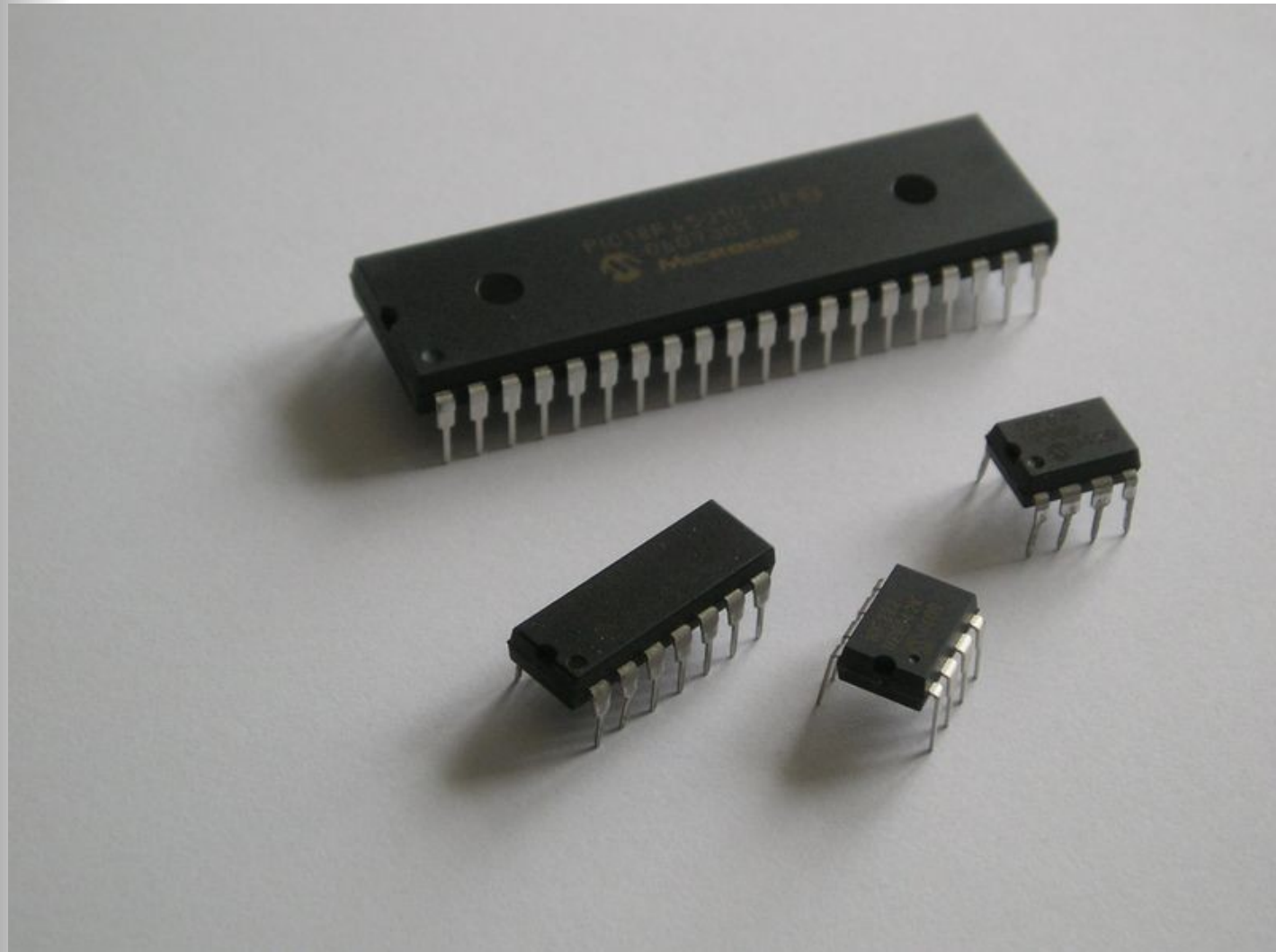


Améliorations envisageables

- Encastrement dans un boîtier
- Affichage de la date
- Affichage progressif de l'heure lors de l'activation de l'alarme



Conclusion



Bibliographie

- Documentation technique du microcontrôleur
- Sites internet :
 - <http://www.ac-orleans-tours.fr/stige/MICROCONTROLEUR/AVR.HTM>
 - <http://www.atmicroprog.com/cours/atmel/atmel.htm>
 - <http://www.jelectronique.com/codevision.php>
 - http://fr.wikipedia.org/wiki/Atmel_AVR

