

UNIVERSITAT POLITÈCNICA DE CATALUNYA

SISTEMES ENCASTATS I UBICS

Entregues laboratoris

Laboratori 2 - Processament de senyal amb ADC

Carlota Catot
Miguel Antunez

Quatrimestre primavera 2020-2021

Índice

1. Codigo	2
2. Resultados	5

1. Código

```
#include "mbed.h"

Serial pc(USBTX, USBRX); // tx, rx
AnalogIn input(p15);
DigitalOut led(LED1);

Timer t;

int main() {

    pc.baud(115200);

    int rate[10];
    unsigned long contador = 0;
    unsigned long tiempoUltimoLatido = 0;
    int P = 520;
    int T = 520;
    int Umbral = 520;
    int amp = 100;
    bool primerLatido = true;
    bool segundoLatido = true;

    int BPM = 0;
    int IBI = 600;
    int Signal = 0;
    bool Pulse = false;
    bool QS = false;
    int timer;

    t.start();
    while (1) {

        timer = t.read_us();
```

Lo primero que se debe hacer es tomar una lectura analógica del sensor de pulso.

A continuación, incrementamos la variable contador.

La variable contador se utiliza para realizar un seguimiento del tiempo.

La variable N ayudará a evitar el ruido.

```
Signal = 1023 * input.read();
contador += 2;
int N = contador - tiempoUltimoLatido;
```

Buscamos el valor más alto y más bajo de la onda.

Las variables P y T mantienen valores máximos y mínimos, respectivamente.

```
if(Signal < Umbral && N > (IBI/5)*3) {
    if (Signal < T) {
```

```

        T = Signal;
    }
}

if(Signal > Umbral && Signal > P) {
    P = Signal;
}

```

La siguiente función buscará un latido. Si la lectura de salida ha pasado el valor de umbral y ha pasado 3/5 del último tiempo entre latidos, entonces la señal de pulso se hará verdadera. Luego, actualizamos la variable "tiempoUltimoLatido" calculando el tiempo desde el último latido.

```

if (N > 250) {
    if ( (Signal > Umbral) && (Pulse == false) && (N > (IBI/5)*3) ) {
        Pulse = true;
        IBI = contador - tiempoUltimoLatido;
        tiempoUltimoLatido = contador;

        if(primerLatido) {
            primerLatido = false;
        }

        if(segundoLatido) {
            segundoLatido = false;
            for(int i=0; i<=9; i++) {
                rate[i] = IBI;
            }
        }

        long runningTotal = 0;

        for(int i=0; i<=8; i++) {
            rate[i] = rate[i+1];
            runningTotal += rate[i];
        }

        rate[9] = IBI;
        runningTotal += rate[9];
        runningTotal /= 10;
        BPM = 60000/runningTotal;
        QS = true;
    }
}

```

Calculamos el latido cuando el valor del pulso es mayor que el valor del umbral. Cuando el valor del pulso es menor que el valor de umbral, asumimos que el pulso ha terminado actualizamos la nueva marca del 50 % para la variable de umbral.

```

if (Signal < Umbral && Pulse == true) {
    Pulse = false;
    amp = P - T;
    Umbral = amp/2 + T;
}

```

```

    P = Umbral;
    T = Umbral;
}

```

Si no encontramos un latido durante unos 2,5 ms, las variables que usamos para calcular el latido del corazón se reinician.

```

if (N > 2500) {
    Umbral = 512;
    P = 512;
    T = 512;
    tiempoUltimoLatido = contador;
    primerLatido = true;
    segundoLatido = true;
}

if (QS == true) {
    QS = false;
    pc.printf("%d \n",BPM);
}

wait_us(2000 - (t.read_us()-timer));
t.reset();
}
}

```

2. Resultados

Con los resultados presentados a continuación podemos ver que la señal tarda en estabilizarse pero una vez estabilizado se marca como que el pulso es de unos 74 pulsaciones, también se puede ver que en un momento se soltó el sensor y por eso tiene variabilidad.

[illegible]

237
237
237
203
185
166
156
154
126
105
87
70
57
54
75
76
79
84
90
95
104
236
236
236
229
229
229
229
229
229
223
223
223
160
140
114
116
116
119
125
94
80
62
61
63
65
95
159
147
146
238
179
148
130
122
117

[illegible]

[illegible]

86
92
88
89
92
93
150
134
129
115
108
102
101
102
98
94
91
94
96
108
109
109
103
100
93
93
97
102
96
91
88
87
84
81
84
81
80
79
79
76
76
77
79
78
79
79
77
74
73
74
75
76
75
76
75
74

74
74
75
75
75
80
79
79
76
76
76
75
76
75
74
74