

Guillaume Risch

Projet terminale SI 2018/2019



Vigie-Chiro
Suivi des
chauves-souris

VIGIENATURE

Un réseau de citoyens qui fait avancer la science

Sommaire

1. *Présentation du projet*
2. *Conception*
3. *Programmation*
4. *Bilan personnel*

Présentation du projet

Nous avons travaillé dans le cadre du Projet Interdisciplinaire sur la création et la conception d'un récepteur enregistreur d'ultrasons, contrôlé à partir d'une carte micro-contrôleur (RaspberryPi). Ce projet a pour but de mettre en œuvre nos compétences acquises lors de cette année, en électronique et en mécanique. La machine en question permet à son utilisateur d'enregistrer les ultrasons des chauves souris et de les comparer afin de les recenser.

Dans le cadre de ce projet, je me suis pour ma part occupé intégralement de la programmation de la carte microcontrôleur. J'ai aussi participé à la conception et à la recherche d'idées afin d'assurer le bon fonctionnement de la machine, et vérifier le respect de notre cahier des charges.

Mes objectifs ont été de programmer de façon efficace la carte RaspberryPi, et de trouver des solutions de conception de la maquette

Conception

Microphone :

Dans ma partie de la conception, je me suis penché sur l'utilisation de microphone afin de pouvoir enregistrer les ultrasons des chauves souris.

L'objectif trouver un système a bas prix afin de concevoir une maquette nous avons donc opté pour le microphone k8118 conçue pour enregistrer les ultrasons des chauves souris.

K8118 Velleman :

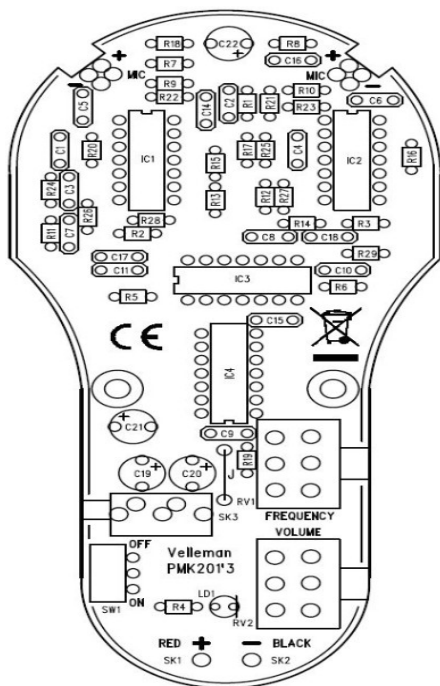
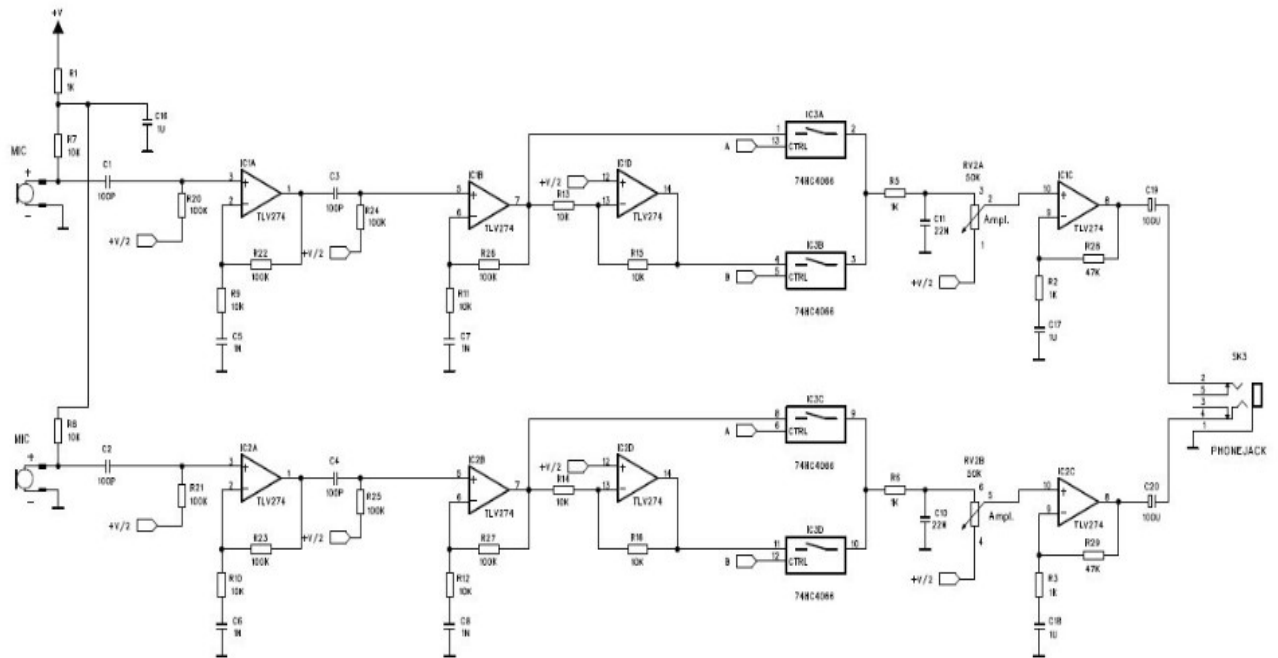
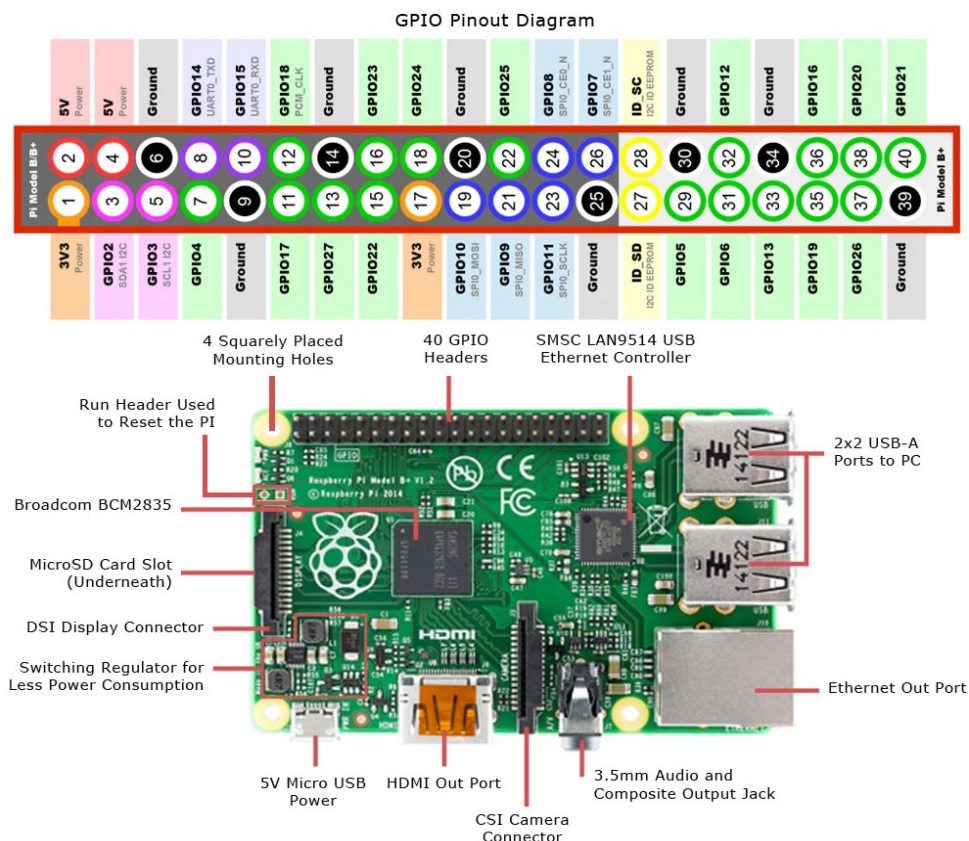


Schéma électronique K8118 Velleman :



RaspberryPi :

En ce qui concerne la carte microcontrôleur, nous avons choisi de travailler sur RaspberryPi. Cette carte permet d'utiliser le langage informatique C, que l'on programme et interprète directement sur la carte, grâce au système d'exploitation dédié Raspbian. Le choix de cette carte a été fait en raison de la puissance de son processeur permettant de répondre à nos attentes, contrairement à la carte micro-contrôleur Arduino pas assez puissante pour obtenir le résultat escompté.



Programmation :

Le processus de programmation a été réalisé en commençant par un algorithme qui est une suite finie d'instructions permettant d'obtenir un résultat ou de résoudre un problème. Chaque instruction correspond à une action simple et bien définie.

Celui ci contre permettant de comparer deux signaux :

Algorithme de caractérisation

Variables :

Signal_Acquis, Signal_Connu : Tableaux de Nombres [1..N] ;
Reconnu : Booléen ;
Cumul : entier dans [1..N] ;
Déviation, Déviation_Locale, MargeLocale, MargeGlobale : Nombres;

Initialisations :

MargeLocale:=40% ;
MargeGlobale:= 10% ;
Reconnu:= Faux
Déviation:= 0 ;

Algorithme :

Tant que (Reconnu = Faux) pour une espèce e faire

 Cumul:=0 ;

 Pour i allant de 1 à N faire

 Déviation_Locale:= |Signal_Acquis[i] – Signal_Connu[i]| ;

 Si (Déviation_Locale>MargeLocale) alors Cumul:=Cumul+1 ;

 Déviation:=Cumul*100/N ;

 Si (Déviation <= MargeGlobale) alors Reconnu := vrai ;
 passer à l'espèce suivante ;

Si (Reconnu = Faux)

 Alors Signal_Acquis n'appartient à aucune des espèces répertoriées ;

Programme :

```
1 // -----
2 // Caracterisation d'un signal acquis relativement à un ensemble de signaux
3 // caracteristiques correspondant à chaque espece repertoriee de chauve-souris
4 //
5 // version 1 : 15/05/19
6 //
7 // -----
8
9 #include <stdio.h>
10 #include <stdlib.h>
11
12 #define NB_ESPECES 10           // nb d'especes connues
13 #define MAX_ECHANTILLONS 1024  // taille max d'un tableau d'echantillonnage
14
15 #define MARGELOCALE 0.4        // marge acceptable d'erreur sur un point
16 // d'echantillonnage
17 #define MARGEGLOBALE 0.1       // marge acceptable d'erreurs cumulees sur
18 // la courbe
19 #define LONGUEUR_CHAINE 32     // taille max d'une chaine de caracteres
20
21 // -----
22 // Structures de donnees
23 // -----
24
25 typedef enum {false, true} bool;
26
27 double SignalAcquis[MAX_ECHANTILLONS];
28 double SignalConnu[NB_ESPECES][MAX_ECHANTILLONS];
29
30 // -----
31 // Fonctions utilitaires
32 // -----
33
34 double Abs (double valeur) {
35     return (valeur<0) ? -valeur : valeur;
36 }
37
38 void Afficher (double SignalAcquis[MAX_ECHANTILLONS]) {
39     for (unsigned int i=0; i<MAX_ECHANTILLONS; ++i)
40         printf("i : %d, echantillon : %f\n", i, SignalAcquis[i]);
41 }
42
43
```



```

43 // -----
44 // Recuperation du signal a tester et de la banque de signaux de reference
45 // -----
46
47 void Init(double SignalAcquis[MAX_ECHANTILLONS]) {
48     char NomFichier[LONGUEUR_CHAINE];
49     FILE * FichierSignalATester;
50
51     printf("Nom du fichier a tester : ");
52     scanf("%s", &NomFichier);
53
54     FichierSignalATester = fopen(NomFichier, "r");
55
56     fread(SignalAcquis, sizeof(double), MAX_ECHANTILLONS, FichierSignalATester);
57
58     fclose(FichierSignalATester);
59 }
60
61 void InitBibliotheque (double SignalConnu[NB_ESPECES][MAX_ECHANTILLONS]) {
62     char NomFichier[LONGUEUR_CHAINE-1];
63     FILE * Bibliotheque;
64
65     printf("Nom du fichier bibliotheque : ");
66     scanf("%s", &NomFichier);
67
68     Bibliotheque = fopen(NomFichier, "r");
69
70     fread(SignalConnu, sizeof(double), NB_ESPECES*MAX_ECHANTILLONS, Bibliotheque);
71
72     fclose(Bibliotheque);
73 }
74
75 // -----
76 // Calcul de la distance entre les courbes
77 // -----
78
79 void Distance (double SignalAcquis[MAX_ECHANTILLONS],
80               double SignalConnu[NB_ESPECES][MAX_ECHANTILLONS]) {
81
82     unsigned int Cumul;
83     double Deviation, DeviationLocale;
84
85     bool SignalReconnu=false;
86
87     for (unsigned int e=0; e<NB_ESPECES; ++e) {
88         Cumul=0;
89         for (unsigned int i=0; i<MAX_ECHANTILLONS; ++i) {
90             DeviationLocale=Abs(SignalAcquis[i]-SignalConnu[e][i]);
91             if (DeviationLocale>MARGELOCALE) ++Cumul;
92         }

```

```

94 // pourcentage de degradation par rapport au signal de reference :
95 Deviation=(double)Cumul/MAX_ECHANTILLONS;
96
97 if (Deviation<=MARGE Globale) {
98     SignalReconnu=true;
99     printf("Signal reconnu appartenant à l'espece %d avec un ecart de %f%%\n",e , Deviation*100);
100 }
101 }
102
103 if (!SignalReconnu) printf("Signal n'appartenant a aucune espece repertoriee\n");
104 }
105
106 // -----
107 // main
108 // -----
109
110 int main (void) {
111     Init(SignalAcquis);
112     InitBibliotheque(SignalConnu);
113
114     // Comparaison du SignalAcquis avec l'ensemble des signaux connus pour
115     // chaque espece
116     Distance(SignalAcquis, SignalConnu);
117
118     return 0;
119 }
120

```

Bilan personnel

Le travail en groupe m'a beaucoup appris. Il m'a d'une part permis de renforcer mes connaissances autour de l'électronique, et m'a d'autre part donné l'occasion de rencontrer un ingénieur qui m'a aidé à mieux comprendre les aspects théoriques de mon projet. Le groupe a tout de même rencontré quelques difficultés liées notamment aux pièces utilisées pour la maquette : certaines de mauvaise facture se sont cassées