

```

1 // ただの音センサー 秋月唯音(3)
2 // 開発環境 MPLAB+HITEC-C
3
4 #include <htc.h>
5
6 #define _XTAL_FREQ 4000000 //delay関数クロック数定義(4MHz)
7 #define TMR0set 6 //TMR0初期値(256-6=250→周期約1ms)
8 #define VOXthreshold 7 //VOX閾値(感度)
9 #define CLAPthreshold 5 //拍手閾値(感度)
10
11 /*環境設定[別途書き込む場合はCONFIG=0x3184]*/
12 __CONFIG( CPD_OFF & //メモリ保護無効
13          CP_OFF & //コードプロテクト無効
14          MCLRE_OFF & //MCLR無効(GP3利用)
15          BOREN_OFF & //BOR監視無効
16          PWRTEN_ON & //電源ONから72ms待機
17          WDT_OFF & //WDT無効
18          FOSC_INTRC10 //内部クロック(4MHz)
19 );
20
21 bit flag; //タイマフラグ
22 int count; //タイマカウント
23 bit CLAPflag; //拍手フラグ
24 int signal; //音声信号取込み値(10bit)
25 int OLDsignal; //音声信号過去値
26 int SIGMA; //音声信号合計値
27 int VOXtime; //VOX動作時間
28 int CLAPtime; //拍手間隔
29
30
31 /*ADC処理: AN1 (VR電圧) */
32 int adcAN1() {
33     ADCON0 = 0b10000101; //AD値は右詰、AN1、VrefはVcc、ADC有効
34     __delay_us(20); //充電待ち
35     GO_nDONE = 1; //AD変換開始
36     while(GO_nDONE); //読取り完了待ち
37     return ((ADRESH << 8) | ADRESL); //10bit化して返す
38 }
39
40 /*ADC処理: AN2 (ECM電圧) */
41 int adcAN2() {
42     ADCON0 = 0b10001001; //AD値は右詰、AN2、VrefはVcc、ADC有効
43     __delay_us(20); //充電待ち
44     GO_nDONE = 1; //AD変換開始
45     while(GO_nDONE); //読取り完了待ち
46     return ((ADRESH << 8) | ADRESL); //10bit化して返す
47 }
48
49 /*拍手モード(二回拍手でトグル) */
50 void CLAP() {
51     /*CLAPモード初期化*/
52     ANSEL = 0b00010110; //8Tosc、AN1,2アナログ
53     GPO = 0; //出力初期化

```

```

54 GP4 = 0; //LED1初期化
55 GP5 = 0; //LED2初期化
56
57 /*スイッチ切り替えるまで無限ループ*/
58 while(GP3 == 0) {
59     OLDsignal = adcAN2(); //マイク信号取込み (1回目)
60     signal = adcAN2(); //マイク信号取込み (2回目)
61
62     if(signal-OLDsignal > CLAPthreshold) { //拍手判定 (高周波検出を狙う)
63         GP4 = 1; //LED1は拍手可能性フラグを兼用
64     }
65
66     /*タイマ周期で処理*/
67     if(flag) {
68         CLAPtime = (adcAN1() << 1) + 100; //半固定抵抗読み取り (2回目の拍手までの時間 [t] を設定)
69
70         if(GP4 == 0) count = 0; //拍手可能性フラグが立たない間はカウントリセット
71
72         if(count >= 10 && count < CLAPtime >> 1) { //無視期間中 (10ms~1/2t) に音検出したら
73             if(signal-OLDsignal > CLAPthreshold) {
74                 GP4 = 0; //拍手可能性フラグダウン
75                 __delay_ms(300); //300msフリーズ (解除直後の再検出を避けるため)
76             }
77         }
78
79         if(count >= CLAPtime >> 1 && count <= CLAPtime) { //2回目受入れ期間 (1/2t~tの間)
80             GP5 = 1; //受け入れ期間中はLED2が点灯
81             if(CLAPflag == 0 && signal-OLDsignal > CLAPthreshold) { //2回目の拍手検出 (2回目以降は無視)
82                 CLAPflag = 1; //2回目が検出されたらフラグアップ
83                 if(GP0) GP0 = 0; //2回目検出で出力をトグル動作
84                 else GP0 = 1;
85             }
86         }
87         else GP5 = 0; //LED2消灯 (受け入れ期間終了)
88
89         if(count > CLAPtime + 100) { //時間tから100ms以上経過したらリセット
90             count = 0; //カウントリセット
91             GP4 = 0; //LED1消灯
92             CLAPflag = 0; //拍手フラグリセット
93         }
94
95         flag = 0; //タイマフラグリセット
96     }
97 }
98 }
99 }
100
101 /*VOX機能 (音で一定時間ON) */
102 void VOX() {
103
104     /*VOXモード初期化*/
105     ANSEL = 0b01010110; ///16Tosc、AN1,2アナログ
106     signal = adcAN2(); //マイク信号取込み (初期値)

```

```

107  GP0 = 0;           //出力初期化
108  GP4 = 1;           //LED1初期化 (反転出力)
109  GP5 = 0;           //LED2初期化
110
111  /*スイッチ切り替えるまで無限ループ*/
112  while (GP3 == 1) {
113      OLDsignal = signal;    //前回マイク信号保存
114      signal = adcAN2();     //マイク信号取込み
115
116      /*マイク信号の差の絶対値を合計*/
117      if (OLDsignal - signal < 0) {
118          SIGMA = SIGMA + signal - OLDsignal;
119      }
120      else SIGMA = SIGMA + OLDsignal - signal;
121
122      /*タイマ周期で処理*/
123      if (flag) {
124
125          VOXtime =(adcAN1 () << 2) + 1;    //半固定抵抗読み取り
126
127          if (SIGMA > VOXthreshold << 2) { //特に大きい音の場合LED2ON
128              GP5 = 1;
129          }
130
131          if (SIGMA > VOXthreshold) {      //音判定
132              GP0 = 1;                     //出力ON
133              GP4 = 0;                     //LED1は反転出力
134              count = 0;                   //カウンタクリア
135          }
136
137          if (count > VOXtime) {          //一定時間経過
138              GP0 = 0;                     //出力OFF
139              GP4 = 1;                     //LED1は反転出力
140              GP5 = 0;                     //LED2もOFF
141              count = 0;                   //カウンタクリア
142          }
143
144          SIGMA = 0;                       //合計値リセット
145          flag = 0;                        //フラグリセット
146      }
147  }
148 }
149 }
150
151 /*タイマ割込処理*/
152 void interrupt TIMERO () {
153     if (TOIF == 1) {                     //TMRO割込発生?
154         TMRO = TMROset;                  //TMRO周期セット
155         TOIF = 0 ;                       //TMRO割込フラグリセット
156         count++;                          //カウントアップ
157         flag = 1;                         //フラグ立てる
158     }
159 }

```

```
160
161
162 /*メイン関数*/
163 void main(){
164
165     /*初期設定*/
166     CMCON = 0b00000111;    //コンパレータ無効（省電力）
167
168     TRISIO = 0b00001110;   //GP1, 2, 3入力
169
170     TMRO = TMR0set;        //TMRO初期化
171     OPTION_REG = 0b00000001; //TMROを内部クロック1:4で使用（基本周期4us）
172     TOIF = 0;              //割込フラグクリア
173     TOIE = 1;              //TMRO割込許可
174     GIE = 1;               //全割込許可
175
176     /*無限ループ*/
177     while(1){
178         if(GP3 == 1)VOX();  //JP1-2ショートでVOXモード
179         else CLAP();        //JP2-3ショートで拍手モード
180     }
181 }
182
```