

```

1  /*****
2  //Name:  main.c
3  //Purpose:  Skeleton project with configuration for ADC, DAC, MCG and PIT
4  //Author:  Ethan Hettwer
5  //Revision:  1.0 15Sept2014 EH Initial Revision
6  //Target:  Freescale K22f
7  *****/
8
9  #include "MK22F51212.h" //Device header
10 #include "MCG.h" //Clock header
11 #include "TimerInt.h" //Timer Interrupt Header
12 #include "ADC.h" //ADC Header
13 #include "DAC.h" //DAC Header
14 #include "PORT.h"
15 #include "useful_func.h"
16 #include <stdio.h>
17 #include <stdlib.h>
18 #include <math.h>
19
20 #define PI 3.14159265
21
22 uint16_t y;
23 uint16_t dac_out;
24 uint8_t i = 0;
25 uint8_t K = 4;
26 //uint8_t a = 12;
27 float fy = 2e2;
28 float Y0 = 0.99;
29 float* yhat;
30 uint8_t size_yhat;
31
32 float Ck[10] = {0.6513, 0.8942, 0.4322, 0.2387, 0.1312, 0.2314, 0.2226, 0.1337, 0.0658, 0.1079};
33 float Ok[10] = {-1.9174, 2.6620, -0.6346, 2.8995, 1.3452, -1.6801, 1.6153, -1.5630, 2.7450, 0.4671};
34
35
36 void PIT0_IRQHandler(void){ //This function is called when the timer interrupt expires
37     //Place Interrupt Service Routine Here
38     // PIT->CHANNEL[0].TFLG = PIT_TFLG_TIF_MASK;
39
40
41     GPIOA->PSOR |= GPIO_PSOR_PTSO(0x1u << 1);
42
43
44     dac_out = (uint16_t) map(yhat[i], -0.5, 3.5, 0, 4095);
45
46     //Output to DAC0
47     DAC0->DAT[0].DATL = DAC_DATL_DATA0( dac_out & 0xFFu );
48     DAC0->DAT[0].DATH = DAC_DATH_DATA1( dac_out >> 8 );
49
50     NVIC_ClearPendingIRQ(PIT0_IRQn); //Clears interrupt flag in NVIC Register
51     PIT->CHANNEL[0].TFLG = PIT_TFLG_TIF_MASK; //Clears interrupt flag in PIT Register
52
53     if(i==(size_yhat-1)) i = 0;
54     else i++;
55
56     GPIOA->PCOR |= GPIO_PCOR_PTCO(0x1u << 1);
57 }
58
59
60 void PORTB_IRQHandler(void){
61
62     GPIOA->PSOR |= GPIO_PSOR_PTSO(0x1u << 1);
63
64     if(K<=4) K++;
65     update_yhat();
66
67     // GPIOA->PCOR |= GPIO_PCOR_PTCO(0x1u << 5);
68
69     NVIC_ClearPendingIRQ(PORTB_IRQn); //Clears interrupt flag in NVIC Register
70     PORTB->PCR[1] |= PORT_PCR_ISF(1u);
71
72     GPIOA->PCOR |= GPIO_PCOR_PTCO(0x1u << 1);

```

```
73 }
74
75 void PORTC_IRQHandler(void) {
76
77     GPIOA->PSOR      |= GPIO_PSOR_PTSC(0x1u << 1);
78
79     if(K<=4) K--;
80     update_yhat();
81
82     // GPIOA->PCOR      |= GPIO_PCOR_PTCO(0x1u << 5);
83
84
85     NVIC_ClearPendingIRQ(PORTB_IRQn);           //Clears interrupt flag in NVIC Register
86     PORTC->PCR[2] |= PORT_PCR_ISF(1u);
87
88     GPIOA->PCOR      |= GPIO_PCOR_PTCO(0x1u << 1);
89 }
90
91
92 void update_yhat() {
93     float Fn = 2*fy*K;
94     float Fs = 12*Fn;
95     float T = 1/Fs;
96     uint32_t ldval = (uint32_t) (60e6 / Fs) - 1u;
97     PIT->CHANNEL[0].LDVAL = PIT_LDVAL_TSV(ldval);
98     size_yhat = 12*K;
99
100     yhat = (float*)malloc(size_yhat * sizeof(float));
101     zeros(yhat, size_yhat);
102
103     float harmonics[K][size_yhat];
104     zeros2d(K, size_yhat, harmonics);
105     generate_harmonics(K, size_yhat, harmonics, fy, K, Ck, Ok);
106
107     add_const_vec(Y0, yhat, size_yhat);
108     for(int k=0; k<K; k++){
109         add_vecs(yhat, harmonics[k], yhat, size_yhat);
110     }
111 }
112
113
114
115
116 int main(void) {
117     MCG_Clock120_Init();
118     ADC_Init();
119     ADC_Calibrate();
120     DAC_Init();
121     TimerInt_Init(4999u);
122     PORT_Init();
123
124     update_yhat();
125
126     while(1) {
127         //Main loop goes here
128     }
129 }
130
131
```