/\*

\* music8.c

\*

\* Created: 10/13/2015 1:47:02 PM

\* Author: haorshao

\*/

#include <avr/io.h>

#include "m\_bus.h"

#include "m\_rf.h"

#include "m\_usb.h"

#include <math.h>

double sin(double \_\_x);

#define channel 1

#define RXADDRESS 0x47

#define PACKET\_LENGTH 3

#define pi 3.1415926

volatile char buffer[3] = {0, 0, 0};

volatile int frequency;

volatile int realFrequency;

volatile int countsine=0;

volatile double realDuration;

volatile double count=0;

volatile int flag;

int main(void)

{

m\_clockdivide(0);//set the system clock frequency as 500KHz

m\_usb\_init();

m\_bus\_init(); // enable mBUS

m\_rf\_open(channel,RXADDRESS,PACKET\_LENGTH); // configure mRF

clear(TCCR1B,CS12);//set prescaler to /64

set(TCCR1B,CS11);

set(TCCR1B,CS10);

//OCR1A=2000000/(64\*frequency);//set the value of OCR1A as 100

//OCR1B=OCR1A/2

set(TCCR1B,WGM13);//(mode 15) UP to OCR1A, PWM mode

set(TCCR1B,WGM12);

set(TCCR1A,WGM11);

set(TCCR1A,WGM10);

set(TCCR1A,COM1B1);//set at OCR1B, clear at rollover

set(TCCR1A,COM1B0);

set(TIMSK1,OCIE1A);//call an interrupt whenever (TCNT1 matches OCR1A)

sei();//enable global interrupt

set(TCCR0B,WGM02); // MODE: up to OCR0A

set(TCCR0A,WGM01); // ^

set(TCCR0A,WGM00); // ^

set(TCCR0A,COM0B1); // clear at OCR0B, set at OCR0A

set(TCCR0A,COM0B0); // ^

set(TCCR0B,CS02); // set prescaler to /1024

clear(TCCR0B,CS01); // ^

set(TCCR0B,CS00); // ^

set(TIMSK0,OCIE0A);

OCR0A=10;

set(DDRB,6);

while(1){

m\_usb\_tx\_int(realFrequency);

m\_usb\_tx\_string("\r\n");

if(flag==1){

frequency= buffer[0]+256\*buffer[1];

//realFrequency=60;

realFrequency=frequency/10;

OCR1A=160000/(64\*realFrequency);

set(DDRB,6);

m\_green(TOGGLE);

TCNT0=0;

count=0;

flag=0;

}else{

m\_green(0);

}

if(count>16\*buffer[2]){

clear(DDRB,6);

count=0;}

if(countsine<100){

OCR1B=((double)OCR1A)/2.0+sin(((double)countsine)/((double)100)\*2\*pi)\*((double)OCR1A)/2.0;//set the value of OCR1B

m\_usb\_tx\_int(OCR1A);

m\_usb\_tx\_int(OCR1B);

m\_usb\_tx\_string("\r\n");

}else{

countsine=0;

}

// OCR1A=520;//set the value of OCR1A as 100

// OCR1B=OCR1A/2;

}

}

ISR(INT2\_vect){

m\_rf\_read(buffer,PACKET\_LENGTH);

flag=1;

//m\_red(TOGGLE);

// frequency= buffer[0]+256\*buffer[1];

// realFrequency=frequency/10;

// OCR1A=2000000/(64\*realFrequency);

// set(DDRB,6);

// m\_green(TOGGLE);

// TCNT0=0;

}

ISR(TIMER1\_COMPA\_vect){

countsine++;

}

ISR(TIMER0\_COMPA\_vect){

count++;

}