```c
#include <avr/io.h>
#include "Digital_Filter.h"
#include "Ring_Buffer.h"

struct Ring_Buffer_F inputs;
struct Ring_Buffer_F outputs;

//a[0] should be entered as the inverse of the coeff, for computation speed
//float b[5] = {0.0940, 0.3759, 0.5639, 0.3759, 0.0940};// cutoff frequency was 250
//float a[5] = {1.0000, 0.0000, 0.4860, 0.0000, 0.0177};
float b[5] = {0.018563010626897, 0.074252042507589, 0.111378063761383, 0.074252042507589, 0.018563010626897}; // cutoff frequency was 150
float a[5] = {1.000000000000000,  -1.570398851228171,  1.275613324983279, -0.484403368335085, 0.076197064610332};
uint8_t i;

/* Initialization */
void digital_filter_init(){
    rb_initialize_F(&inputs);
    rb_initialize_F(&outputs);

    for(i = 0; i <= order + 1; i++){
        rb_push_front_F(&inputs, 0);
        rb_push_front_F(&outputs, 0);
    }

    return;
}

float filterValue(float newInput){
    rb_pop_back_F(&inputs);
    rb_pop_back_F(&outputs);
    rb_push_front_F(&inputs, newInput);
    float newOutput = 0;
    for(i=0; i < order + 1; i++){
        newOutput += b[i]*rb_get_F(&inputs,i);
        if (i>0){
            newOutput -= a[i]*rb_get_F(&outputs,i-1);
        }
    }
    newOutput *= a[0];
    rb_push_front_F(&outputs, newOutput);
    return newOutput;
}
```