```c
/*
 * File:   message.c
 * Author: redxef
 *
 * Created on 18 March 2016, 09:25
 */


#include <xc.h>
#include "message.h"
#include "powerspy.h"

int8_t readpos;
char reading;

void initMessaging()
{
        buffpos = -1;
        readpos = 0;
        reading = 0;
}

int8_t charAvailable()
{
        if (buffpos < readpos) {
                return RECEIVEBUFF_SIZE - readpos + buffpos;
        }
        return buffpos - readpos;
}

char readNext()
{
        char ret = receive_buff[readpos++];
        if (readpos == RECEIVEBUFF_SIZE)
                readpos = 0;
        return ret;
}

char packageStarted()
{
        int8_t i;
        for (i = 0; i < RECEIVEBUFF_SIZE; i++) {
                if (receive_buff[i] == START_OF_TEXT)
                        return 1;
        }
        return 0;
}

char packageFinished()
{
        int8_t i;
        if (!packageStarted())
                return 0;
        for (i = 1; i < RECEIVEBUFF_SIZE; i++)
                if (receive_buff[i] == END_OF_TEXT)
                        return 1;
```

```
 57         return 0;
 58 }
 59
 60 char getType()
 61 {
 62         int8_t i;
 63         for (i = 0; i < RECEIVEBUFF_SIZE; i++) {
 64                 if (receive_buff[i] == START_OF_TEXT) {
 65                         return receive_buff[i + 1];
 66                 }
 67         }
 68         return NONE;
 69 }
 70
 71 void seekFront()
 72 {
 73         int8_t i;
 74         for (i = 0; i < RECEIVEBUFF_SIZE; i++) {
 75                 if (receive_buff[i] == START_OF_TEXT) {
 76                         readpos = i + 2;
 77                 }
 78         }
 79 }
 80
 81 void clear()
 82 {
 83         int8_t i;
 84         for (i = 0; i < RECEIVEBUFF_SIZE; i++) {
 85                 receive_buff[i] = 0;
 86         }
 87         reading = 0;
 88         readpos = 0;
 89 }
 90
 91 int8_t readInt8()
 92 {
 93         int8_t res = 0;
 94
 95         seekFront();
 96         res = (int8_t) readNext();
 97         clear();
 98
 99         return res;
100 }
101
102 int16_t readInt16()
103 {
104         int16_t res = 0;
105
106         seekFront();
107         res = readNext();
108         res <<= 8;
109         res |= readNext();
110         clear();
111
112         return res;
113 }
```

```
114
115 int24_t readInt24()
116 {
117         int24_t res = 0;
118
119         seekFront();
120         res = readNext();
121         res <<= 8;
122         res |= readNext();
123         res <<= 8;
124         res |= readNext();
125         clear();
126
127         return res;
128 }
129
130 int32_t readInt32()
131 {
132         int32_t res = 0;
133
134         seekFront();
135         res = readNext();
136         res <<= 8;
137         res |= readNext();
138         res <<= 8;
139         res |= readNext();
140         res <<= 8;
141         res |= readNext();
142         clear();
143
144         return res;
145 }
146
147 float readFloat()
148 {
149         int24_t i = readInt24();
150         float *res = (float *) & i;
151         return *res;
152 }
153
154 void readString(char **c)
155 {
156
157 }
158
159 void _sendchar_(char c)
160 {
161         TXREG = c;
162         while (!TRMT);
163         __delay_ms(1);
164 }
165
166 void sendInt8(int8_t i)
167 {
168         _sendchar_(START_OF_TEXT);
169         _sendchar_(INT8);
170         _sendchar_(i);
```

```
171 }
172
173 void sendInt16(int16_t i)
174 {
175         _sendchar_(START_OF_TEXT);
176         _sendchar_(INT16);
177         _sendchar_((char) (i >> 8 & 0xff));
178         _sendchar_((char) (i & 0xff));
179 }
180
181 void sendInt24(int24_t i)
182 {
183         _sendchar_(START_OF_TEXT);
184         _sendchar_(INT24);
185         _sendchar_((char) (i >> 16 & 0xff));
186         _sendchar_((char) (i >> 8 & 0xff));
187         _sendchar_((char) (i & 0xff));
188 }
189
190 void sendInt32(int32_t i)
191 {
192         _sendchar_(START_OF_TEXT);
193         _sendchar_(INT32);
194         _sendchar_((char) (i >> 24 & 0xff));
195         _sendchar_((char) (i >> 16 & 0xff));
196         _sendchar_((char) (i >> 8 & 0xff));
197         _sendchar_((char) (i & 0xff));
198 }
199
200 void sendUInt8(uint8_t i)
201 {
202         _sendchar_(START_OF_TEXT);
203         _sendchar_(UINT8);
204         _sendchar_(i);
205 }
206
207 void sendUInt16(uint16_t i)
208 {
209         _sendchar_(START_OF_TEXT);
210         _sendchar_(UINT16);
211         _sendchar_((char) (i >> 8 & 0xff));
212         _sendchar_((char) (i & 0xff));
213 }
214
215 void sendUInt24(uint24_t i)
216 {
217         _sendchar_(START_OF_TEXT);
218         _sendchar_(UINT24);
219         _sendchar_((char) (i >> 16 & 0xff));
220         _sendchar_((char) (i >> 8 & 0xff));
221         _sendchar_((char) (i & 0xff));
222 }
223
224 void sendUInt32(uint32_t i)
225 {
226         _sendchar_(START_OF_TEXT);
227         _sendchar_(UINT32);
```

```
228         _sendchar_((char) (i >> 24 & 0xff));
229         _sendchar_((char) (i >> 16 & 0xff));
230         _sendchar_((char) (i >> 8 & 0xff));
231         _sendchar_((char) (i & 0xff));
232 }
233
234 void sendFloat(float f)
235 {
236         uint24_t *ptr = (uint24_t *) & f;
237         _sendchar_(START_OF_TEXT);
238         _sendchar_(FLOAT);
239         _sendchar_((char) (*ptr >> 16 & 0xff));
240         _sendchar_((char) (*ptr >> 8 & 0xff));
241         _sendchar_((char) (*ptr & 0xff));
242 }
243
244 void sendString(char *val)
245 {
246         _sendchar_(START_OF_TEXT);
247         _sendchar_(STRING);
248         while (*val) {
249                 _sendchar_(*val);
250                 val++;
251         }
252         _sendchar_(END_OF_TEXT);
253 }
```