C:/microchip/harmony/v2_06/apps/PROJ/2230_TubePitotDeporte_v1.0.0/firmware/src/RN4678_driver.c

```
3 * File: RN4678_driver.c
4 * Author: M.Ricchieri
6 * Inspired by the code of S. Giuseppe
7
   * Created on 12. avril 2023
8 :
9 * This code uses USART with FIFO
10 */
11
12
13 //----// Includes
14 #include <stdbool.h>
15 #include <stdint.h>
16 #include "RN4678 driver.h"
17 #include "app.h"
18
19
20 //----// Constants
21 // Commands
22 #define CMD_MODE_ENTER "$$$\r"
23 #define CMD MODE EXIT "---\r"
23 #define CMD MODE EXIT
24 #define CMD BLE DISCOV EN "Q,0\r"
25 // The module is able to connect, but is undiscoverable in Bluetooth Classic
26 #define CMD_BT_DISCOV_DIS "Q,2\r"
27 #define CMD BLE ONLY
                            "SG,1\r"
28 #define CMD BT CLASSIC ONLY "SG,2\r"
29 #define CMD_PREFIX_SUFIX "SO,<,>\r"
30 #define CMD_REBOOT_DEVICE "R,1\r"
31 #define CMD_BITMAP "SQ,8000\r"
32 #define CMD_SCAN_DURATION "SL,01\r"
33
34 // Answers
35 #define CMD_MODE ANSWER "CMD>"
36 #define CMD EXIT ANSWER "END\r\n"
37 #define CMD_POS_ANSWER "AOK\r\nCMD> "
38 #define CMD_NEG_ANSWER "ERR\r\nCMD> "
39 //#define CMD_REBOOT_ANSWER "<REBOOT>"
40 #define CMD_REBOOT_ANSWER "Rebooting\r\n"
41
42 // Device name
43 #define DEVICE NAME
                           "SN, TubePitotDeporte v1.0.0\r"
44
45
46
47 //-----// init RN4678
48 bool init_RN4678(void){
49
     bool initIsDone = 1;
50
51
   //Resets the module for a reboot
52
    RESET_BLEOff();
53
54
      inv_imu_sleep_ms(1000);
    RESET_BLEOn();
55
56
     inv_imu_sleep_ms(2000);
57
     appData.isBluetoothInCommandMode = true;
58
    // Enters in command mode
59
60
    initIsDone = sendCMD RN4678(CMD MODE ENTER, sizeof(CMD MODE ENTER), CMD MODE ANSWER,
61
            sizeof(CMD MODE ANSWER));
62
      // Sets the name of the device
    initIsDone &= sendCMD RN4678(DEVICE NAME, sizeof(DEVICE NAME), CMD POS ANSWER,
63
64
             sizeof(CMD_POS_ANSWER));
65
      // Sets the Bluetooth mode in Classic
     initIsDone &= sendCMD_RN4678(CMD_BT_CLASSIC_ONLY, sizeof(CMD_BT_CLASSIC_ONLY), CMD_POS_ANSWER,
66
            sizeof(CMD POS ANSWER));
    // Sets the prefix and the sufix of status
68
69
      initIsDone &= sendCMD RN4678(CMD PREFIX SUFIX, sizeof(CMD PREFIX SUFIX), CMD POS ANSWER,
70
             sizeof(CMD POS ANSWER));
71
    // Sets the scan duration to 10 seconds
72
     initIsDone &= sendCMD_RN4678(CMD_SCAN_DURATION, sizeof(CMD_SCAN_DURATION), CMD_POS_ANSWER,
73
             sizeof(CMD_POS_ANSWER));
      // Sets the RN4678 into Fast mode
74
```

1.1 of 3 2023.06.15 00:41:27

C:/microchip/harmony/v2_06/apps/PROJ/2230_TubePitotDeporte_v1.0.0/firmware/src/RN4678_driver.c

```
initIsDone &= sendCMD_RN4678(CMD_BITMAP, sizeof(CMD_BITMAP), CMD_POS_ANSWER,
 76
              sizeof(CMD POS ANSWER));
77
       // Lauches a reboot command
      initIsDone &= sendCMD RN4678(CMD REBOOT DEVICE, sizeof(CMD REBOOT DEVICE), CMD REBOOT ANSWER,
78
79
              sizeof(CMD_REBOOT_ANSWER));
80
81
      inv_imu_sleep_ms(2000);
82
8.3
      // Flag is discoverable true
84
      appData.isBluetoothDiscoverable = true;
8.5
      appData.isBluetoothInCommandMode = false;
87
88
      return initIsDone;
89 }
90
91 //-----// turnOffDiscoverBT
92 // For the moment, this function isn't used. The Fast mode affects the detection
93 // of the "$$$\r" message and it is impossible to inter in command mode when the
94 // data mode is enable.
95 bool turnOffDiscoverBT(void){
96
      appData.isBluetoothInCommandMode = true;
97
98 // // Enters in command mode
      sendCMD_RN4678(CMD_MODE_ENTER, sizeof(CMD_MODE_ENTER), CMD_MODE_ANSWER,
99
100
             sizeof(CMD MODE ANSWER));
101
     // Turn off the discoverable mode of the module
102
    sendCMD_RN4678(CMD_BT_DISCOV_DIS, sizeof(CMD_BT_DISCOV_DIS), CMD_POS_ANSWER,
103
             sizeof(CMD POS ANSWER));
104
      // Exits command mode
105
     sendCMD RN4678(CMD MODE EXIT, sizeof(CMD MODE EXIT), CMD EXIT ANSWER,
106
             sizeof(CMD_EXIT_ANSWER));
107
      appData.isBluetoothInCommandMode = false;
108
109
110
      return 1;
111 }
112
113
114 //-----// sendCMD RN4678
115 bool sendCMD_RN4678(char* pArrayToSend, size_t arraySize, char* pArrayExpected,
116
          size t answerSize) {
117
118
     int8_t a_answer[20];
119
120
      // Save data in TX FIFO
121
     putStringInFifo(&usartFifoTx, arraySize, pArrayToSend);
122
      // Enable USART TX interrupt
123
      PLIB INT SourceEnable(INT ID 0, INT SOURCE USART 1 TRANSMIT);
124
125
126
          // If the number of new char in FIFO is the same as the answer size
127
          if(getReadSize(&usartFifoRx) >= answerSize - 1){
128
129
              // Reads the answere received
130
              getStringFromFifo(&usartFifoRx, &a answer[0]);
131
          1
132
133
      }while((strstr((char*)a_answer, pArrayExpected) == NULL));
134
          //if(strstr((char*)a_answer, pArrayExpected) != NULL) isInitDone = 1;
135
136
      //}while(isInitDone != 1);
137
138
      clearInt8Array(sizeof(a_answer), &a_answer[0]);
139
140
      return 1;
141 }
142
143
144 //-----// qetUsartData
145 void getUsartData(int8_t* pArrayToModify){
146
147
      do{
```

2.1 of 3 2023.06.15 00:41:27

C:/microchip/harmony/v2_06/apps/PROJ/2230_TubePitotDeporte_v1.0.0/firmware/src/RN4678_driver.c

```
149
         getStringFromFifo(&usartFifoRx, &pArrayToModify[0]);
150
151
      }while(getReadSize(&usartFifoRx));
152 }
153
154
155 //-----// sendData RN4678
156 void sendData_RN4678(int8_t* pArrayToSend){
157
158
     int cursor = 0;
159
    // Does until character '\r' is sent
160
161
     do{
         // Wait for the Transmit buffer to be empty.
162
163
        if(!PLIB_USART_TransmitterBufferIsFull(USART_ID_1)){
164
165
             // Sends all data of the array
            PLIB_USART_TransmitterByteSend(USART_ID_1, pArrayToSend[cursor]);
166
167
       }
168
     }while(pArrayToSend[cursor-1] != '\r');
169
170 }
171
172
173 //-----// readStatus
174 void readStatus(char *pArrayStatus){
175
176
     int cursor = 0;
177
   while(PLIB_USART_ReceiverDataIsAvailable(USART_ID_1)){
179
    \ensuremath{//} Reads and saves the characters received in an array
       pArrayStatus[cursor] = PLIB_USART_ReceiverByteReceive(USART_ID_1);
// Increments the cursor value
180
181
       cursor++;
182
    }
183
184 }
185
186
187 //-----// clearInt8Array
188 void clearInt8Array(size_t arraySize, int8_t* arrayToClear){
189
190
191
     for (i = 0; i < arraySize; i++){
192
193
194
       arrayToClear[i] = NULL;
195
    }
196 }
```

3.1 of 3 2023.06.15 00:41:27