

com_config.h

[Go to the documentation of this file.](#)

```
1 #ifndef _COM_CONFIG_H_ 2 #define _COM_CONFIG_H_ 3 4 #include <vector> 5 #include <string> 6 #include <
pthread.h> 7 #include "ipc/jsobject.h" 8 10 class ComConfig 11 { 12 private: 13 // disable copy
constructor and assign operator 14 ComConfig(const ComConfig&); 15 ComConfig& operator=(const ComConfig&);
16 #if __cplusplus >= 201103L // since C++11 also forbid usage of move 17 ComConfig(const ComConfig&&); 18
ComConfig& operator=(ComConfig&&); 19 #endif 20 21 // mutex protecting members 22 pthread_mutex_t
m_mutex; 23 26 std::string m_profile; 27 std::vector<std::string> m_whitelist; 28 std::string
m_bt_address; 29 std::string m_bt_device; 30 std::string m_comla_address; 31 std::string m_comla_device;
32 bool m_comla_btclient_reconnect_idle; 33 unsigned m_comla_mode; 34 bool m_comla_btserver_pairing_idle;
35 std::string m_ble_address; 36 std::string m_ble_device; 37 std::string m_bt_own_device_name; 38
unsigned m_bt_num_paired; 42 int m_idx; 43 std::vector<unsigned> m_idx_list; 45 ComConfig(); 46 virtual
~ComConfig(); 47 48 public: 49 // operation modes for COM1A profile 50 enum ComlaMode 51 { 52
Comla_Usb = 0, // COM1A USB (default) 53 Comla_BTClient = 1, // COM1A BT-client 54 Comla_BTServer = 2 //
COM1A BT-server 55 }; 56 59 static ComConfig& getInstance(); 60 63 inline void lock() 64 { 65
pthread_mutex_lock(&m_mutex); 66 } 68 inline void unlock() 69 { 70 pthread_mutex_unlock(&m_mutex); 71
} 74 inline pthread_mutex_t &getMutex() 75 { 76 return m_mutex; 77 } 78 83 inline std::string&
profile() 84 { 85 return m_profile; // written/changed by select_com_profile() 86 } 89 inline const
std::vector<std::string>& whitelist() 90 { 91 return m_whitelist; 92 } 95 inline std::string&
bt_address() 96 { 97 return m_bt_address; // written/changes by bt_pan_menu() 98 } 101 inline
std::string& bt_device() 102 { 103 return m_bt_device; // written/changes by bt_pan_menu() 104 } 107
inline std::string& comla_address() 108 { 109 return m_comla_address; // written/changes by comla_menu()
110 } 113 inline std::string& comla_device() 114 { 115 return m_comla_device; // written/changes by
comla_menu() 116 } 119 inline bool& comla_btclient_reconnect_idle() 120 { 121 return
m_comla_btclient_reconnect_idle; // written/changed by comla_menu() 122 } 125 inline unsigned& comla_mode()
126 { 127 return m_comla_mode; // written/changed by comla_menu() 128 } 131 inline bool&
comla_btserver_pairing_idle() 132 { 133 return m_comla_btserver_pairing_idle; // written/changed by
comla_menu() 134 } 137 inline std::string& ble_address() 138 { 139 return m_ble_address; //
written/changed by bt_pairing_submenu() 140 } 143 inline std::string& ble_device() 144 { 145 return
m_ble_device; // written/changed by bt_pairing_submenu() 146 } 149 inline const std::string&
bt_own_device_name() 150 { 151 return m_bt_own_device_name; // written/changed by
activateConnectionProfile() 152 } 155 inline unsigned bt_num_paired() 156 { 157 return m_bt_num_paired;
158 } 159 162 inline int& idx() 163 { 164 return m_idx; // written/changed by select_com_profile() and
displayCOMMenu() 165 } 168 inline std::vector<unsigned>& idx_list() 169 { 170 return m_idx_list; //
written/changed by select_com_profile() 171 } 172 174 void clear(); 175 182 void read(bool
*storage_needed = 0); 183 185 void store(); 186 190 static std::string getComlaModeName(unsigned mode);
191 200 static bool load(const std::string &com_if_file, vfiipc::JSObject &res, bool *storage_needed = 0);
201 203 enum MultiConnectMode 204 { 205 MCM_Default = 0, 206 MCM_Enabled = 1, 207 MCM_Disabled = 2
208 }; 214 static enum MultiConnectMode getMultiConnectMode(); 215 }; 216 217 #endif // _COM_CONFIG_H_
218
```