

Iridium Modem Library

1.0

Generated by Doxygen 1.8.12

Contents

- 1 Class Index** **1**
 - 1.1 Class List 1

- 2 File Index** **3**
 - 2.1 File List 3

- 3 Class Documentation** **5**
 - 3.1 IR9602 Class Reference 5
 - 3.1.1 Member Enumeration Documentation 6
 - 3.1.1.1 InterfaceMode 6
 - 3.1.1.2 MainState 6
 - 3.1.2 Member Function Documentation 6
 - 3.1.2.1 disable() 6
 - 3.1.2.2 enable() 6
 - 3.1.2.3 getRxBuffer() 6
 - 3.1.2.4 getTotalReceived() 7
 - 3.1.2.5 getTotalSent() 7
 - 3.1.2.6 getTxBuffer() 7
 - 3.1.2.7 init() 7
 - 3.1.2.8 manualStartSession() 8
 - 3.1.2.9 newReceived() 8
 - 3.1.2.10 process() 8
 - 3.1.2.11 setCycleMode() 8
 - 3.1.2.12 setSendAttemptPeriod() 9

3.1.2.13	transmitSuccess()	9
3.1.3	Member Data Documentation	9
3.1.3.1	txBuffer	9
3.2	SerialPacket Class Reference	9
3.2.1	Detailed Description	10
3.2.2	Constructor & Destructor Documentation	10
3.2.2.1	SerialPacket()	10
3.2.3	Member Function Documentation	10
3.2.3.1	getPacketPtr()	10
3.2.3.2	listen()	10
3.2.4	Member Data Documentation	10
3.2.4.1	newData	10
4	File Documentation	11
4.1	DebugMacros.h File Reference	11
4.1.1	Macro Definition Documentation	11
4.1.1.1	DEBUG_PRINT	11
4.1.1.2	DEBUG_PRINTLN	11
4.2	elapsedMillis.h File Reference	11
4.3	IR9602.cpp File Reference	11
4.4	IR9602.h File Reference	12
4.4.1	Macro Definition Documentation	12
4.4.1.1	IR9602_DEBUG	12
4.4.1.2	IR_DBG_PRINT	12
4.4.1.3	IR_DBG_PRINTLN	12
4.5	main.cpp File Reference	12
4.5.1	Function Documentation	13
4.5.1.1	blinkLED()	13
4.5.1.2	control()	13
4.5.1.3	loop()	13
4.5.1.4	setup()	13

4.5.1.5	SignalCounter()	13
4.5.2	Variable Documentation	14
4.5.2.1	command	14
4.5.2.2	iridium	14
4.5.2.3	timer	14
4.6	SerialPacket.cpp File Reference	14
4.7	SerialPacket.h File Reference	14
4.8	settings.h File Reference	14
4.8.1	Macro Definition Documentation	15
4.8.1.1	DEBUG	15
4.8.1.2	GPS_SERIAL	15
4.8.1.3	IRIDIUM_SERIAL	15
4.8.1.4	XBEE_SERIAL	15
4.8.2	Variable Documentation	15
4.8.2.1	I2C_SLAVE_ADDR	15
4.8.2.2	P_GPS_PV	16
4.8.2.3	P_GPS_RESET	16
4.8.2.4	P_GPS_RXD1	16
4.8.2.5	P_GPS_TXD1	16
4.8.2.6	P_IR_NET_AVAIL	16
4.8.2.7	P_IR_PWR_ON	16
4.8.2.8	P_IR_RX	16
4.8.2.9	P_IR_TX	16
4.8.2.10	P_LED_LOCK	16
4.8.2.11	P_LED_STAT1	16
4.8.2.12	P_LED_STAT2	17
4.8.2.13	P_LED_TRANS	17
4.8.2.14	P_SERVO_CNTRL	17
4.8.2.15	P_XB_ASSOCIATED	17
4.8.2.16	P_XB_DIN	17
4.8.2.17	P_XB_DOUT	17
4.8.2.18	P_XB_SLEEP	17

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

IR9602	5
SerialPacket Class for serial packet	9

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

DebugMacros.h	11
elapsedMillis.h	11
IR9602.cpp	11
IR9602.h	12
main.cpp	12
SerialPacket.cpp	14
SerialPacket.h	14
settings.h	14

Chapter 3

Class Documentation

3.1 IR9602 Class Reference

```
#include <IR9602.h>
```

Public Types

- enum `InterfaceMode` { `IM_AUTO_SEND`, `IM_MANUAL_SEND` }
- enum `MainState` { `S_OFF`, `S_IDLE`, `S_PUTTING_TX_BUFFER`, `S_GETTING_TX_BUFFER_ACK`, `S_SESSION_IN_PROGRESS`, `S_GETTING_RX_BUFFER`, `S_COMM_ERROR` }

Public Member Functions

- void `init` (HardwareSerial &irUART, uint8_t pinEnable, uint8_t pinNetAvail)
Initializes hardware pins and class variables and sets the baud rate.
- void `process` ()
Processes packets coming in from modem.
- void `enable` ()
Enables or starts the iridium modem.
- void `disable` ()
Disables or stops the iridium modem.
- void `setSendAttemptPeriod` (uint8_t seconds)
Sets the send attempt period.
- void `setCycleMode` (`InterfaceMode` userMode)
Sets the cycle mode.
- bool `manualStartSession` ()
Manually starts a data session.
- bool `newReceived` ()
Returns if there's been a new message.
- bool `transmitSuccess` ()
Returns if a transmission was successful.
- uint16_t `getTotalSent` ()
Gets the total sent.
- uint16_t `getTotalReceived` ()
Gets the total received.
- bool `getTxBuffer` (String *bufferPtr)
Gets the transmit buffer.
- bool `getRxBuffer` (String *bufferPtr)
Gets the receive buffer.

Public Attributes

- String `txBuffer`

3.1.1 Member Enumeration Documentation

3.1.1.1 InterfaceMode

enum `IR9602::InterfaceMode`

Enumerator

IM_AUTO_SEND	
IM_MANUAL_SEND	

3.1.1.2 MainState

enum `IR9602::MainState`

Enumerator

S_OFF	
S_IDLE	
S_PUTTING_TX_BUFFER	
S_GETTING_TX_BUFFER_ACK	
S_SESSION_IN_PROGRESS	
S_GETTING_RX_BUFFER	
S_COMM_ERROR	

3.1.2 Member Function Documentation

3.1.2.1 disable()

```
void IR9602::disable ( )
```

Disables or stops the iridium modem.

3.1.2.2 enable()

```
void IR9602::enable ( )
```

Enables or starts the iridium modem.

3.1.2.3 getRxBuffer()

```
bool IR9602::getRxBuffer (
    String * bufferPtr )
```

Gets the receive buffer.

Parameters

<i>bufferPtr</i>	The buffer pointer
------------------	--------------------

Returns

Sets the recieved buffer and returns success/failure.

3.1.2.4 getTotalReceived()

```
uint16_t IR9602::getTotalReceived ( )
```

Gets the total received.

Returns

The total received.

3.1.2.5 getTotalSent()

```
uint16_t IR9602::getTotalSent ( )
```

Gets the total sent.

Returns

The total sent.

3.1.2.6 getTxBuffer()

```
bool IR9602::getTxBuffer (
    String * bufferPtr )
```

Gets the transmit buffer.

Parameters

<i>bufferPtr</i>	The buffer pointer
------------------	--------------------

Returns

The transmit buffer.

3.1.2.7 init()

```
void IR9602::init (
    HardwareSerial & irUART,
```

```
uint8_t pinEnable,
uint8_t pinNetAvail )
```

Initializes hardware pins and class variables and sets the baud rate.

Parameters

	<i>irUART</i>	Serial pin to iridium modem
in	<i>pinEnable</i>	Pin to enable iridium modem
in	<i>pinNetAvail</i>	The pin net avail

3.1.2.8 manualStartSession()

```
bool IR9602::manualStartSession ( )
```

Manually starts a data session.

Returns

returns true if the session is manual, false otherwise.

3.1.2.9 newReceived()

```
bool IR9602::newReceived ( )
```

Returns if there's been a new message.

Returns

returns true if a new message was recieved, flase otherwise.

3.1.2.10 process()

```
void IR9602::process ( )
```

Processes packets coming in from modem.

3.1.2.11 setCycleMode()

```
void IR9602::setCycleMode (
    IR9602::InterfaceMode mode )
```

Sets the cycle mode.

Parameters

in	<i>mode</i>	The mode
----	-------------	----------

3.1.2.12 setSendAttemptPeriod()

```
void IR9602::setSendAttemptPeriod (
    uint8_t seconds )
```

Sets the send attempt period.

Parameters

in	<i>seconds</i>	The seconds fo period (millisencond -> second conversion handled).
----	----------------	--

3.1.2.13 transmitSuccess()

```
bool IR9602::transmitSuccess ( )
```

Returns if a transmission was successfull.

Returns

returns true if success, false otherwise.

3.1.3 Member Data Documentation

3.1.3.1 txBuffer

```
String IR9602::txBuffer
```

The documentation for this class was generated from the following files:

- [IR9602.h](#)
- [IR9602.cpp](#)

3.2 SerialPacket Class Reference

Class for serial packet.

```
#include <SerialPacket.h>
```

Public Member Functions

- [SerialPacket](#) ()
Constructor.
- void [listen](#) ()
Listens for data on Serial, usually AT commands.
- const String * [getPacketPtr](#) ()
Gets the packet pointer.

Public Attributes

- bool [newData](#)

3.2.1 Detailed Description

Class for serial packet.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 SerialPacket()

```
SerialPacket::SerialPacket ( )
```

Constructor.

3.2.3 Member Function Documentation

3.2.3.1 getPacketPtr()

```
const String * SerialPacket::getPacketPtr ( )
```

Gets the packet pointer.

Returns

The packet pointer.

3.2.3.2 listen()

```
void SerialPacket::listen ( )
```

Listens for data on Serial, usually AT commands.

3.2.4 Member Data Documentation

3.2.4.1 newData

```
bool SerialPacket::newData
```

The documentation for this class was generated from the following files:

- [SerialPacket.h](#)
- [SerialPacket.cpp](#)

Chapter 4

File Documentation

4.1 DebugMacros.h File Reference

```
#include "settings.h"
```

Macros

- #define `DEBUG_PRINT(x)` `Serial.print(x)`
- #define `DEBUG_PRINTLN(x)` `Serial.println(x)`

4.1.1 Macro Definition Documentation

4.1.1.1 `DEBUG_PRINT`

```
#define DEBUG_PRINT(  
    x ) Serial.print(x)
```

4.1.1.2 `DEBUG_PRINTLN`

```
#define DEBUG_PRINTLN(  
    x ) Serial.println(x)
```

4.2 elapsedMillis.h File Reference

4.3 IR9602.cpp File Reference

```
#include "IR9602.h"
```

4.4 IR9602.h File Reference

```
#include <Arduino.h>
#include "elapsedMillis.h"
#include "DebugMacros.h"
#include "settings.h"
```

Classes

- class [IR9602](#)

Macros

- #define [IR9602_DEBUG](#) 1
- #define [IR_DBG_PRINT](#)(x) Serial.print(x)
- #define [IR_DBG_PRINTLN](#)(x) Serial.println(x)

4.4.1 Macro Definition Documentation

4.4.1.1 IR9602_DEBUG

```
#define IR9602_DEBUG 1
```

4.4.1.2 IR_DBG_PRINT

```
#define IR_DBG_PRINT(  
    x ) Serial.print(x)
```

4.4.1.3 IR_DBG_PRINTLN

```
#define IR_DBG_PRINTLN(  
    x ) Serial.println(x)
```

4.5 main.cpp File Reference

```
#include <Arduino.h>
#include "settings.h"
#include "DebugMacros.h"
#include "IR9602.h"
#include "SerialPacket.h"
```

Functions

- void `blinkLED` ()
Blinks the on board LED.
- void `control` ()
Handles data coming in on Serial, usually AT commands.
- char `SignalCounter` ()
Counts the signal quality.
- void `setup` ()
- void `loop` ()

Variables

- `IR9602` iridium
- `SerialPacket` command
- int `timer`

4.5.1 Function Documentation

4.5.1.1 `blinkLED()`

```
void blinkLED ( )
```

Blinks the on board LED.

4.5.1.2 `control()`

```
void control ( )
```

Handles data coming in on Serial, usually AT commands.

4.5.1.3 `loop()`

```
void loop ( )
```

4.5.1.4 `setup()`

```
void setup ( )
```

4.5.1.5 `SignalCounter()`

```
char SignalCounter ( )
```

Counts the signal quality.

Returns

Signal quality value.

4.5.2 Variable Documentation

4.5.2.1 command

`SerialPacket` command

4.5.2.2 iridium

`IR9602` iridium

4.5.2.3 timer

`int` timer

4.6 SerialPacket.cpp File Reference

```
#include "SerialPacket.h"
```

4.7 SerialPacket.h File Reference

```
#include <Arduino.h>
#include <string.h>
#include <HardwareSerial.h>
#include "elapsedMillis.h"
```

Classes

- class `SerialPacket`
Class for serial packet.

4.8 settings.h File Reference

Macros

- `#define` `DEBUG` 1
- `#define` `IRIDIUM_SERIAL` Serial1
- `#define` `XBEE_SERIAL` Serial2
- `#define` `GPS_SERIAL` Serial3

Variables

- const int [P_LED_LOCK](#) = 14
- const int [P_LED_TRANS](#) = 15
- const int [P_LED_STAT1](#) = 16
- const int [P_LED_STAT2](#) = 17
- const int [P_XB_DOUT](#) = 9
- const int [P_XB_DIN](#) = 10
- const int [P_XB_SLEEP](#) = 11
- const int [P_XB_ASSOCIATED](#) = 12
- const int [P_GPS_RESET](#) = 6
- const int [P_GPS_TXD1](#) = 7
- const int [P_GPS_RXD1](#) = 8
- const int [P_GPS_PV](#) = 28
- const int [P_IR_RX](#) = 0
- const int [P_IR_TX](#) = 1
- const int [P_IR_PWR_ON](#) = 2
- const int [P_IR_NET_AVAIL](#) = 24
- const int [P_SERVO_CNTRL](#) = 23
- const int [I2C_SLAVE_ADDR](#) = 2

4.8.1 Macro Definition Documentation

4.8.1.1 DEBUG

```
#define DEBUG 1
```

4.8.1.2 GPS_SERIAL

```
#define GPS_SERIAL Serial3
```

4.8.1.3 IRIDIUM_SERIAL

```
#define IRIDIUM_SERIAL Serial1
```

4.8.1.4 XBEE_SERIAL

```
#define XBEE_SERIAL Serial2
```

4.8.2 Variable Documentation

4.8.2.1 I2C_SLAVE_ADDR

```
const int I2C_SLAVE_ADDR = 2
```

4.8.2.2 P_GPS_PV

```
const int P_GPS_PV = 28
```

4.8.2.3 P_GPS_RESET

```
const int P_GPS_RESET = 6
```

4.8.2.4 P_GPS_RXD1

```
const int P_GPS_RXD1 = 8
```

4.8.2.5 P_GPS_TXD1

```
const int P_GPS_TXD1 = 7
```

4.8.2.6 P_IR_NET_AVAIL

```
const int P_IR_NET_AVAIL = 24
```

4.8.2.7 P_IR_PWR_ON

```
const int P_IR_PWR_ON = 2
```

4.8.2.8 P_IR_RX

```
const int P_IR_RX = 0
```

4.8.2.9 P_IR_TX

```
const int P_IR_TX = 1
```

4.8.2.10 P_LED_LOCK

```
const int P_LED_LOCK = 14
```

4.8.2.11 P_LED_STAT1

```
const int P_LED_STAT1 = 16
```

4.8.2.12 P_LED_STAT2

```
const int P_LED_STAT2 = 17
```

4.8.2.13 P_LED_TRANS

```
const int P_LED_TRANS = 15
```

4.8.2.14 P_SERVO_CNTRL

```
const int P_SERVO_CNTRL = 23
```

4.8.2.15 P_XB_ASSOCIATED

```
const int P_XB_ASSOCIATED = 12
```

4.8.2.16 P_XB_DIN

```
const int P_XB_DIN = 10
```

4.8.2.17 P_XB_DOUT

```
const int P_XB_DOUT = 9
```

4.8.2.18 P_XB_SLEEP

```
const int P_XB_SLEEP = 11
```


Index

- blinkLED
 - main.cpp, 13
- command
 - main.cpp, 14
- control
 - main.cpp, 13
- DEBUG_PRINTLN
 - DebugMacros.h, 11
- DEBUG_PRINT
 - DebugMacros.h, 11
- DEBUG
 - settings.h, 15
- DebugMacros.h, 11
 - DEBUG_PRINTLN, 11
 - DEBUG_PRINT, 11
- disable
 - IR9602, 6
- elapsedMillis.h, 11
- enable
 - IR9602, 6
- GPS_SERIAL
 - settings.h, 15
- getPacketPtr
 - SerialPacket, 10
- getRxBuffer
 - IR9602, 6
- getTotalReceived
 - IR9602, 7
- getTotalSent
 - IR9602, 7
- getTxBuffer
 - IR9602, 7
- I2C_SLAVE_ADDR
 - settings.h, 15
- IR9602, 5
 - disable, 6
 - enable, 6
 - getRxBuffer, 6
 - getTotalReceived, 7
 - getTotalSent, 7
 - getTxBuffer, 7
 - init, 7
 - InterfaceMode, 6
 - MainState, 6
 - manualStartSession, 8
 - newReceived, 8
 - process, 8
 - setCycleMode, 8
 - setSendAttemptPeriod, 9
 - transmitSuccess, 9
 - txBuffer, 9
- IR9602.cpp, 11
- IR9602.h, 12
 - IR9602_DEBUG, 12
 - IR_DBG_PRINTLN, 12
 - IR_DBG_PRINT, 12
- IR9602_DEBUG
 - IR9602.h, 12
- IR_DBG_PRINTLN
 - IR9602.h, 12
- IR_DBG_PRINT
 - IR9602.h, 12
- IRIDIUM_SERIAL
 - settings.h, 15
- init
 - IR9602, 7
- InterfaceMode
 - IR9602, 6
- iridium
 - main.cpp, 14
- listen
 - SerialPacket, 10
- loop
 - main.cpp, 13
- main.cpp, 12
 - blinkLED, 13
 - command, 14
 - control, 13
 - iridium, 14
 - loop, 13
 - setup, 13
 - SignalCounter, 13
 - timer, 14
- MainState
 - IR9602, 6
- manualStartSession
 - IR9602, 8
- newData
 - SerialPacket, 10
- newReceived
 - IR9602, 8
- P_GPS_PV

- settings.h, 15
- P_GPS_RESET
 - settings.h, 16
- P_GPS_RXD1
 - settings.h, 16
- P_GPS_TXD1
 - settings.h, 16
- P_IR_NET_AVAIL
 - settings.h, 16
- P_IR_PWR_ON
 - settings.h, 16
- P_IR_RX
 - settings.h, 16
- P_IR_TX
 - settings.h, 16
- P_LED_LOCK
 - settings.h, 16
- P_LED_STAT1
 - settings.h, 16
- P_LED_STAT2
 - settings.h, 16
- P_LED_TRANS
 - settings.h, 17
- P_SERVO_CNTRL
 - settings.h, 17
- P_XB_ASSOCIATED
 - settings.h, 17
- P_XB_DIN
 - settings.h, 17
- P_XB_DOUT
 - settings.h, 17
- P_XB_SLEEP
 - settings.h, 17
- process
 - IR9602, 8
- SerialPacket, 9
 - getPacketPtr, 10
 - listen, 10
 - newData, 10
 - SerialPacket, 10
- SerialPacket.cpp, 14
- SerialPacket.h, 14
- setCycleMode
 - IR9602, 8
- setSendAttemptPeriod
 - IR9602, 9
- settings.h, 14
 - DEBUG, 15
 - GPS_SERIAL, 15
 - I2C_SLAVE_ADDR, 15
 - IRIDIUM_SERIAL, 15
 - P_GPS_PV, 15
 - P_GPS_RESET, 16
 - P_GPS_RXD1, 16
 - P_GPS_TXD1, 16
 - P_IR_NET_AVAIL, 16
 - P_IR_PWR_ON, 16
 - P_IR_RX, 16
 - P_IR_TX, 16
 - P_LED_LOCK, 16
 - P_LED_STAT1, 16
 - P_LED_STAT2, 16
 - P_LED_TRANS, 17
 - P_SERVO_CNTRL, 17
 - P_XB_ASSOCIATED, 17
 - P_XB_DIN, 17
 - P_XB_DOUT, 17
 - P_XB_SLEEP, 17
 - XBEE_SERIAL, 15
- setup
 - main.cpp, 13
- SignalCounter
 - main.cpp, 13
- timer
 - main.cpp, 14
- transmitSuccess
 - IR9602, 9
- txBuffer
 - IR9602, 9
- XBEE_SERIAL
 - settings.h, 15