

usb

Version 2. Copyright (c) 2023 Terrortronics / Bradley Elenbaas. All rights reserved.

Generated by Doxygen 1.9.7

1 usb	1
1.1 Attribution	1
1.2 Intellectual Property	1
1.3 License	1
1.4 Description	1
1.5 Pins	2
1.6 Resources	2
1.7 WARNINGS	2
2 File Index	3
2.1 File List	3
3 File Documentation	5
3.1 usb.ino File Reference	5
3.1.1 Enumeration Type Documentation	5
3.1.1.1 messages	5
3.1.2 Function Documentation	6
3.1.2.1 loop()	6
3.1.3 WARNING	6
3.1.3.1 reset()	6
3.1.3.2 setup()	6
3.1.3.3 start()	6
3.1.4 Variable Documentation	6
3.1.4.1 CONFIG	6
3.1.4.2 counter	7
3.1.4.3 COUNTER_MAX	7
3.1.4.4 incomingMessage	7
3.1.4.5 RESET_MESSAGE	7
3.1.4.6 SERIAL_BAUD_RATE	7
3.1.4.7 START_MESSAGE	7
Index	9

Chapter 1

usb

1.1 Attribution

- Title: USB file.
- Author: Terrortronics / Bradley Elenbaas (mr.elenbaas@gmail.com)
- Version: 2
- Date: September 25, 2023.

1.2 Intellectual Property

- Copyright (c) 2023 Bradley Elenbaas. All rights reserved.

1.3 License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.4 Description

The USB library.

1.5 Pins

- empty

1.6 Resources

- https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/using-arduino-ide?gclid=CjwKCAjwh4ObBhAzEiwAHzZYU9fWXPNZrw8oHU1V8xxeILSFGz1O_08SyU8i6gU0pNtaYe↵BZTgg0exoCMEQQAvD_BwE
- <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>
- <https://arduinogetstarted.com/faq/how-to-reset-arduino-by-programming>

1.7 WARNINGS

- DO NO USE THIS EXAMPLE (<https://reference.arduino.cc/reference/en/language/functions/communication/serial/read-string/>) (Reason #1): The readString function may cause memory leaks.
- DO NO USE THIS EXAMPLE (<https://reference.arduino.cc/reference/en/language/functions/communication/serial/serial-blocking-code/>) (Reason #2): The Serial blocking code breaks on Unity Standalone Windows builds.
- DO NOT USE the pySerial.write() call inside any PyGame code, and instead use the Thread library to separate their respective blocks with flags to move data back and forth.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

usb.ino	5
-----------------------------------	---

Chapter 3

File Documentation

3.1 usb.ino File Reference

Enumerations

- enum `messages` { `startMessage` = START_MESSAGE , `resetMessage` = RESET_MESSAGE }

Functions

- void `setup` ()
- void `loop` ()
- void `reset` ()
- void `start` ()

Variables

- const int `SERIAL_BAUD_RATE` = 9600
- const int `START_MESSAGE` = 48
- const int `RESET_MESSAGE` = 49
- int `incomingMessage`
- const String `CONFIG` = "type:config,filename:joystick_manager,function:reset,delimiters: cm0;\n"
- int `COUNTER_MAX` = 4
- int `counter`

3.1.1 Enumeration Type Documentation

3.1.1.1 messages

enum `messages`

An enum for parsing an incoming byte. Starts with ASCII 0 (48) and goes up from there.

Enumerator

startMessage	
resetMessage	

3.1.2 Function Documentation

3.1.2.1 loop()

```
void loop ( )
```

The main function.

3.1.3 WARNING

- Remember to consume the incoming bytes.
 - The error does not occur when using the usb.c or usb.py files.
 - The error does occur when reading/writing in a PyGame application.

3.1.3.1 reset()

```
void reset ( )
```

A stub function to be populated on a per-project basis.

3.1.3.2 setup()

```
void setup ( )
```

Set up Serial.

3.1.3.3 start()

```
void start ( )
```

Outputs the [CONFIG](#) string.

3.1.4 Variable Documentation

3.1.4.1 CONFIG

```
const String CONFIG = "type:config,filename:joystick_manager,function:reset,delimiters:  cm0;\n"
```

The config to be parsed by a PC application.

3.1.4.2 counter

```
int counter
```

The counter to be output.

3.1.4.3 COUNTER_MAX

```
int COUNTER_MAX = 4
```

The maximum count for [counter](#).

3.1.4.4 incomingMessage

```
int incomingMessage
```

A single incoming byte.

3.1.4.5 RESET_MESSAGE

```
const int RESET_MESSAGE = 49
```

Associated with the [reset\(\)](#) function.

3.1.4.6 SERIAL_BAUD_RATE

```
const int SERIAL_BAUD_RATE = 9600
```

Baud rate (pair with PC).

3.1.4.7 START_MESSAGE

```
const int START_MESSAGE = 48
```

Associated with the [start\(\)](#) function.

Index

- CONFIG
 - [usb.ino, 6](#)
- counter
 - [usb.ino, 6](#)
- COUNTER_MAX
 - [usb.ino, 7](#)
- incomingMessage
 - [usb.ino, 7](#)
- loop
 - [usb.ino, 6](#)
- messages
 - [usb.ino, 5](#)
- reset
 - [usb.ino, 6](#)
- RESET_MESSAGE
 - [usb.ino, 7](#)
- resetMessage
 - [usb.ino, 6](#)
- SERIAL_BAUD_RATE
 - [usb.ino, 7](#)
- setup
 - [usb.ino, 6](#)
- start
 - [usb.ino, 6](#)
- START_MESSAGE
 - [usb.ino, 7](#)
- startMessage
 - [usb.ino, 6](#)
- [usb, 1](#)
- [usb.ino, 5](#)
 - [CONFIG, 6](#)
 - [counter, 6](#)
 - [COUNTER_MAX, 7](#)
 - [incomingMessage, 7](#)
 - [loop, 6](#)
 - [messages, 5](#)
 - [reset, 6](#)
 - [RESET_MESSAGE, 7](#)
 - [resetMessage, 6](#)
 - [SERIAL_BAUD_RATE, 7](#)
 - [setup, 6](#)
 - [start, 6](#)
 - [START_MESSAGE, 7](#)
 - [startMessage, 6](#)