

keyword.txt dossier /lib

keyword.txt

```
#FUNCTIONS COLOR          #D35400 - ORANGE          KEYWORD1
#FUNCTIONS COLOR          #D35400 - ORANGE          KEYWORD2
#STRUCTURE COLORS        #728E00 - GREEN          KEYWORD3
#VARIABLES COLOR         #00979C - BLUE           LITERAL1

# LITERAL1 specifies constants

HIGH    LITERAL1    Constants    RESERVED_WORD_2
LOW    LITERAL1    Constants    RESERVED_WORD_2
INPUT  LITERAL1    Constants    RESERVED_WORD_2
INPUT_PULLUP    LITERAL1    Constants    RESERVED_WORD_2
OUTPUT LITERAL1    Constants    RESERVED_WORD_2
DEC    LITERAL1    Serial_Print    RESERVED_WORD_2
BIN    LITERAL1    Serial_Print    RESERVED_WORD_2
HEX    LITERAL1    Serial_Print    RESERVED_WORD_2
OCT    LITERAL1    Serial_Print    RESERVED_WORD_2
PI     LITERAL1    RESERVED_WORD_2
HALF_PI LITERAL1    RESERVED_WORD_2
TWO_PI LITERAL1    RESERVED_WORD_2
LSBFIRST    LITERAL1    ShiftOut    RESERVED_WORD_2
MSBFIRST    LITERAL1    ShiftOut    RESERVED_WORD_2
CHANGE     LITERAL1    AttachInterrupt    RESERVED_WORD_2
FALLING    LITERAL1    AttachInterrupt    RESERVED_WORD_2
RISING     LITERAL1    AttachInterrupt    RESERVED_WORD_2
DEFAULT    LITERAL1    AnalogReference    RESERVED_WORD_2
EXTERNAL   LITERAL1    AnalogReference    RESERVED_WORD_2
INTERNAL   LITERAL1    AnalogReference    RESERVED_WORD_2
INTERNAL1V1 LITERAL1    AnalogReference    RESERVED_WORD_2
INTERNAL2V56    LITERAL1    AnalogReference    RESERVED_WORD_2
LED_BUILTIN LITERAL1    Constants    RESERVED_WORD_2
LED_BUILTIN_RX LITERAL1    Constants    RESERVED_WORD_2
LED_BUILTIN_TX LITERAL1    Constants    RESERVED_WORD_2

DIGITAL_MESSAGE LITERAL1    Constants    RESERVED_WORD_2
FIRMATA_STRING  LITERAL1    Constants    RESERVED_WORD_2
ANALOG_MESSAGE  LITERAL1    Constants    RESERVED_WORD_2
REPORT_DIGITAL  LITERAL1    Constants    RESERVED_WORD_2
REPORT_ANALOG   LITERAL1    Constants    RESERVED_WORD_2
SET_PIN_MODE    LITERAL1    Constants    RESERVED_WORD_2
SYSTEM_RESET    LITERAL1    Constants    RESERVED_WORD_2
SYSEX_START    LITERAL1    Constants    RESERVED_WORD_2

auto    LITERAL1    RESERVED_WORD_2
int8_t  LITERAL1    RESERVED_WORD_2
int16_t LITERAL1    RESERVED_WORD_2
```

int32_t	LITERAL1		RESERVED_WORD_2
int64_t	LITERAL1		RESERVED_WORD_2
uint8_t	LITERAL1		RESERVED_WORD_2
uint16_t	LITERAL1		RESERVED_WORD_2
uint32_t	LITERAL1		RESERVED_WORD_2
uint64_t	LITERAL1		RESERVED_WORD_2
char16_t	LITERAL1		RESERVED_WORD_2
char32_t	LITERAL1		RESERVED_WORD_2
operator	LITERAL1		RESERVED_WORD_2
enum	LITERAL1		RESERVED_WORD_2
delete	LITERAL1		RESERVED_WORD_2
bool	LITERAL1		RESERVED_WORD_2
boolean	LITERAL1	BooleanVariables	RESERVED_WORD_2
byte	LITERAL1	Byte	RESERVED_WORD_2
char	LITERAL1	Char	RESERVED_WORD_2
const	LITERAL1	Const	RESERVED_WORD_2
false	LITERAL1	Constants	LITERAL_BOOLEAN
float	LITERAL1	Float	RESERVED_WORD_2
double	LITERAL1		RESERVED_WORD_2
null	LITERAL1		RESERVED_WORD_2
NULL	LITERAL1		RESERVED_WORD_2
int	LITERAL1	Int	RESERVED_WORD_2
long	LITERAL1	Long	RESERVED_WORD_2
new	LITERAL1		RESERVED_WORD_2
private	LITERAL1		RESERVED_WORD_2
protected	LITERAL1		RESERVED_WORD_2
public	LITERAL1		RESERVED_WORD_2
short	LITERAL1		RESERVED_WORD_2
signed	LITERAL1		RESERVED_WORD_2
static	LITERAL1	Static	RESERVED_WORD_2
volatile	LITERAL1	Volatile	RESERVED_WORD_2
String	LITERAL1	String	RESERVED_WORD_2
void	LITERAL1	Void	RESERVED_WORD_2
true	LITERAL1		LITERAL_BOOLEAN
unsigned	LITERAL1		RESERVED_WORD_2
word	LITERAL1		RESERVED_WORD_2
array	LITERAL1	Constants	RESERVED_WORD_2
sizeof	LITERAL1	Constants	RESERVED_WORD_2
dynamic_cast	LITERAL1	Constants	RESERVED_WORD_2
typedef	LITERAL1	Constants	RESERVED_WORD_2
const_cast	LITERAL1	Constants	RESERVED_WORD_2
struct	LITERAL1	Constants	RESERVED_WORD_2
static_cast	LITERAL1	Constants	RESERVED_WORD_2
union	LITERAL1	Constants	RESERVED_WORD_2
friend	LITERAL1	Constants	RESERVED_WORD_2
extern	LITERAL1	Constants	RESERVED_WORD_2
class	LITERAL1	Constants	RESERVED_WORD_2
reinterpret_cast	LITERAL1	Constants	RESERVED_WORD_2

```

register      LITERAL1      Constants      RESERVED_WORD_2
explicit     LITERAL1      Constants      RESERVED_WORD_2
inline      LITERAL1      Constants      RESERVED_WORD_2
_Bool       LITERAL1      Constants      RESERVED_WORD_2
complex     LITERAL1      Constants      RESERVED_WORD_2
_Complex    LITERAL1      Constants      RESERVED_WORD_2
_Imaginary  LITERAL1      Constants      RESERVED_WORD_2
atomic_bool LITERAL1      Constants      RESERVED_WORD_2
atomic_char LITERAL1      Constants      RESERVED_WORD_2
atomic_schar LITERAL1     Constants      RESERVED_WORD_2
atomic_uchar LITERAL1     Constants      RESERVED_WORD_2
atomic_short LITERAL1     Constants      RESERVED_WORD_2
atomic_ushort LITERAL1    Constants      RESERVED_WORD_2
atomic_int  LITERAL1     Constants      RESERVED_WORD_2
atomic_uint LITERAL1     Constants      RESERVED_WORD_2
atomic_long LITERAL1     Constants      RESERVED_WORD_2
atomic_ulong LITERAL1    Constants      RESERVED_WORD_2
atomic_llong LITERAL1    Constants      RESERVED_WORD_2
atomic_ullong LITERAL1    Constants      RESERVED_WORD_2
virtual     LITERAL1     Constants      RESERVED_WORD_2
PROGMEM     LITERAL1     Constants      RESERVED_WORD_2

```

KEYWORD2 specifies methods and functions

```

abs KEYWORD2      Abs
acos      KEYWORD2      ACos
acosf     KEYWORD2
asin      KEYWORD2      ASin
asinf     KEYWORD2
atan      KEYWORD2      ATan
atan2     KEYWORD2      ATan2
atan2f    KEYWORD2
atanf     KEYWORD2
cbrt     KEYWORD2
cbrtf    KEYWORD2
ceil     KEYWORD2      Ceil
ceilf    KEYWORD2
constrain KEYWORD2      Constrain
copysign KEYWORD2
copysignf KEYWORD2
cos      KEYWORD2      Cos
cosf     KEYWORD2
cosh     KEYWORD2
coshf    KEYWORD2
degrees  KEYWORD2
exp      KEYWORD2      Exp
expf     KEYWORD2
fabs     KEYWORD2
fabsf    KEYWORD2
fdim     KEYWORD2
fdimf    KEYWORD2

```

```
floor KEYWORD2 Floor
floorf KEYWORD2
fma KEYWORD2
fmaf KEYWORD2
fmax KEYWORD2
fmaxf KEYWORD2
fmin KEYWORD2
fminf KEYWORD2
fmod KEYWORD2
fmodf KEYWORD2
hypot KEYWORD2
hypotf KEYWORD2
isfinite KEYWORD2
isinf KEYWORD2
isnan KEYWORD2
ldexp KEYWORD2
ldexpf KEYWORD2
log KEYWORD2 Log
log10 KEYWORD2
log10f KEYWORD2
logf KEYWORD2
lrint KEYWORD2
lrintf KEYWORD2
lround KEYWORD2
lroundf KEYWORD2
map KEYWORD2 Map
max KEYWORD2 Max
min KEYWORD2 Min
pow KEYWORD2 Pow
powf KEYWORD2
radians KEYWORD2
random KEYWORD2 Random
randomSeed KEYWORD2 RandomSeed
round KEYWORD2
roundf KEYWORD2
signbit KEYWORD2
sin KEYWORD2 Sin
sinf KEYWORD2
sinh KEYWORD2
sinhf KEYWORD2
sq KEYWORD2 Sq
sqrt KEYWORD2 Sqrt
sqrtf KEYWORD2
tan KEYWORD2 Tan
tanf KEYWORD2
tanh KEYWORD2
tanhf KEYWORD2
trunc KEYWORD2
truncf KEYWORD2
```

```
bitRead KEYWORD2 BitRead
bitWrite KEYWORD2 BitWrite
bitSet KEYWORD2 BitSet
bitClear KEYWORD2 BitClear
bit KEYWORD2 Bit
highByte KEYWORD2 HighByte
lowByte KEYWORD2 LowByte

analogReference KEYWORD2 AnalogReference
analogRead KEYWORD2 AnalogRead
analogReadResolution KEYWORD2 AnalogReadResolution
analogWrite KEYWORD2 AnalogWrite
analogWriteResolution KEYWORD2 AnalogWriteResolution
attachInterrupt KEYWORD2 AttachInterrupt
detachInterrupt KEYWORD2 DetachInterrupt
digitalPinToInterrupt KEYWORD2 DigitalPinToInterrupt
delay KEYWORD2 Delay
delayMicroseconds KEYWORD2 DelayMicroseconds
digitalWrite KEYWORD2 DigitalWrite
digitalRead KEYWORD2 DigitalRead
interrupts KEYWORD2
millis KEYWORD2 Millis
micros KEYWORD2 Micros
noInterrupts KEYWORD2 NoInterrupts
noTone KEYWORD2 NoTone
pinMode KEYWORD2 PinMode
pulseIn KEYWORD2 PulseIn
pulseInLong KEYWORD2 PulseInLong
shiftIn KEYWORD2 ShiftIn
shiftOut KEYWORD2 ShiftOut
tone KEYWORD2 Tone
yield KEYWORD2 Yield

Stream KEYWORD2
Serial KEYWORD1 Serial DATA_TYPE
Serial1 KEYWORD1 Serial DATA_TYPE
Serial2 KEYWORD1 Serial DATA_TYPE
Serial3 KEYWORD1 Serial DATA_TYPE
SerialUSB KEYWORD1 Serial DATA_TYPE
begin KEYWORD2 Serial_Begin
end KEYWORD2 Serial_End
peek KEYWORD2 Serial_Peek
read KEYWORD2 Serial_Read
print KEYWORD2 Serial_Print
println KEYWORD2 Serial_Println
available KEYWORD2 Serial_Available
availableForWrite KEYWORD2
flush KEYWORD2 Serial_Flush
setTimeout KEYWORD2
find KEYWORD2
```

```
findUntil    KEYWORD2
parseInt     KEYWORD2
parseFloat   KEYWORD2
readBytes    KEYWORD2
readBytesUntil KEYWORD2
readString   KEYWORD2
readStringUntil KEYWORD2
trim         KEYWORD2
toUpperCase  KEYWORD2
toLowerCase  KEYWORD2
charAt       KEYWORD2
compareTo    KEYWORD2
concat       KEYWORD2
endsWith     KEYWORD2
startsWith   KEYWORD2
equals       KEYWORD2
equalsIgnoreCase KEYWORD2
getBytes     KEYWORD2
indexOf      KEYWORD2
lastIndexOf  KEYWORD2
length       KEYWORD2
replace      KEYWORD2
setCharAt    KEYWORD2
substring    KEYWORD2
toArray      KEYWORD2
toInt        KEYWORD2

Keyboard     KEYWORD1      DATA_TYPE
Mouse        KEYWORD1      DATA_TYPE
press        KEYWORD2
release      KEYWORD2
releaseAll   KEYWORD2
accept       KEYWORD2
click        KEYWORD2
move         KEYWORD2
isPressed    KEYWORD2

isAlphaNumeric KEYWORD2
isAlpha         KEYWORD2
isAscii         KEYWORD2
isWhitespace    KEYWORD2
isControl       KEYWORD2
isDigit         KEYWORD2
isGraph         KEYWORD2
isLowerCase     KEYWORD2
isPrintable     KEYWORD2
isPunct         KEYWORD2
isSpace         KEYWORD2
isUpperCase     KEYWORD2
```

```
isHexadecimalDigit KEYWORD2

# KEYWORD3 specifies structures

break KEYWORD3 Break RESERVED_WORD
case KEYWORD3 SwitchCase RESERVED_WORD
override KEYWORD3 RESERVED_WORD
final KEYWORD3 RESERVED_WORD
continue KEYWORD3 Continue RESERVED_WORD
default KEYWORD3 SwitchCase RESERVED_WORD
do KEYWORD3 DoWhile RESERVED_WORD
else KEYWORD3 Else RESERVED_WORD
for KEYWORD3 For RESERVED_WORD
if KEYWORD3 If RESERVED_WORD
return KEYWORD3 Return RESERVED_WORD
goto KEYWORD3 RESERVED_WORD

switch KEYWORD3 SwitchCase RESERVED_WORD
throw KEYWORD3 RESERVED_WORD
try KEYWORD3 RESERVED_WORD
while KEYWORD3 While RESERVED_WORD

setup KEYWORD3 Setup RESERVED_WORD
loop KEYWORD3 Loop RESERVED_WORD
export KEYWORD3 RESERVED_WORD

not KEYWORD3 If RESERVED_WORD
or KEYWORD3 If RESERVED_WORD
and KEYWORD3 If RESERVED_WORD
xor KEYWORD3 If RESERVED_WORD

# operators aren't highlighted, but may have documentation

+= IncrementCompound
+ Arithmetic
[] arrayaccess
= assign
& BitwiseAnd
| BitwiseAnd
^ BitwiseAnd
~ BitwiseXorNot
,
// Comments
?:
{} Braces
-- Increment
/ Arithmetic
/* Comments
. dot
== If
< If
```

```
<=      If
++      Increment
!=      If
<<     Bitshift
>       If
>=     If
&&     Boolean
!       Boolean
||      Boolean
-       Arithmetic
%       Modulo
*       Arithmetic
()      parentheses
>>     Bitshift
;       SemiColon
-=     DecrementCompound

#include  KEYWORD3      PREPROCESSOR
#define  KEYWORD3      PREPROCESSOR
#elif   KEYWORD3      PREPROCESSOR
#else   KEYWORD3      PREPROCESSOR
#error  KEYWORD3      PREPROCESSOR
#if     KEYWORD3      PREPROCESSOR
#ifdef  KEYWORD3      PREPROCESSOR
#ifndef KEYWORD3      PREPROCESSOR
#pragma KEYWORD3      PREPROCESSOR
#warning KEYWORD3      PREPROCESSOR
```

From: <https://www.magenealogie.chanterie37.fr/www/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link: <https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=start:arduino:ide:keyword&rev=1664442536>

Last update: 2023/01/27 16:08

