

Sortiermaschine

Generated by Doxygen 1.8.17

1 Main Page	1
1.1 Einleitung	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 AnimatableLcd Class Reference	9
5.1.1 Member Function Documentation	11
5.1.1.1 init()	11
5.1.1.2 print()	11
5.1.1.3 printCentered()	11
5.1.1.4 printPretty()	11
5.1.1.5 setAnimation()	12
5.1.1.6 update()	12
5.1.2 Member Data Documentation	12
5.1.2.1 animString	12
5.1.2.2 doAnimation	12
5.2 AnimString Class Reference	13
5.2.1 Constructor & Destructor Documentation	14
5.2.1.1 ~AnimString()	14
5.2.2 Member Function Documentation	14
5.2.2.1 init()	14
5.2.2.2 LcdString()	14
5.2.2.3 run()	15
5.2.3 Member Data Documentation	15
5.2.3.1 animStart	15
5.2.3.2 lastRefresh	15
5.2.3.3 stepDuration	15
5.3 ButtonHandler Class Reference	15
5.3.1 Constructor & Destructor Documentation	16
5.3.1.1 ButtonHandler() [1/2]	16
5.3.1.2 ButtonHandler() [2/2]	16
5.3.2 Member Function Documentation	16
5.3.2.1 update()	16
5.3.3 Member Data Documentation	16
5.3.3.1 isPressed	16
5.3.3.2 onclick	16

5.3.3.3 pin	17
5.4 Callable Struct Reference	17
5.4.1 Constructor & Destructor Documentation	17
5.4.1.1 ~Callable()	17
5.4.2 Member Function Documentation	18
5.4.2.1 isDone()	18
5.4.2.2 run()	18
5.5 CallHandler Class Reference	18
5.5.1 Detailed Description	19
5.5.2 Constructor & Destructor Documentation	19
5.5.2.1 ~ CallHandler()	19
5.5.3 Member Function Documentation	19
5.5.3.1 deleteCalls()	19
5.5.3.2 setCalls()	19
5.5.3.3 update()	20
5.5.4 Member Data Documentation	20
5.5.4.1 callPtrs	20
5.5.4.2 callsSet	20
5.5.4.3 currCallPtr	20
5.5.4.4 lastCallPtr	20
5.5.4.5 lastCallT	20
5.5.4.6 running	21
5.6 CustomServo Class Reference	21
5.6.1 Detailed Description	22
5.6.2 Member Function Documentation	22
5.6.2.1 isDone()	23
5.6.2.2 setSpeed()	23
5.6.2.3 start()	23
5.6.2.4 startMove()	23
5.6.2.5 stop()	23
5.6.2.6 updatePos()	23
5.6.2.7 write() [1/2]	23
5.6.2.8 write() [2/2]	24
5.6.2.9 writeDirect()	24
5.6.3 Member Data Documentation	24
5.6.3.1 done	24
5.6.3.2 servo	24
5.6.3.3 speed	24
5.6.3.4 startAngle	25
5.6.3.5 startTime	25
5.6.3.6 targetAngle	25
5.7 FuncCall Struct Reference	25

5.7.1 Constructor & Destructor Documentation	26
5.7.1.1 FuncCall() [1/2]	26
5.7.1.2 FuncCall() [2/2]	26
5.7.1.3 ~FuncCall()	26
5.7.2 Member Function Documentation	26
5.7.2.1 isDone()	27
5.7.2.2 run()	27
5.7.3 Member Data Documentation	27
5.7.3.1 _isDone	27
5.7.3.2 call	27
5.8 LcdDotAnim Class Reference	28
5.8.1 Constructor & Destructor Documentation	29
5.8.1.1 LcdDotAnim()	29
5.8.2 Member Function Documentation	29
5.8.2.1 init()	29
5.8.2.2 update()	29
5.9 LcdLoadingAnim Class Reference	30
5.9.1 Member Function Documentation	31
5.9.1.1 init()	31
5.9.1.2 update()	31
5.10 LcdString Struct Reference	31
5.10.1 Constructor & Destructor Documentation	32
5.10.1.1 LcdString()	33
5.10.1.2 ~LcdString()	33
5.10.2 Member Function Documentation	33
5.10.2.1 isDone()	33
5.10.2.2 run()	33
5.10.2.3 update()	33
5.10.3 Member Data Documentation	34
5.10.3.1 callStart	34
5.10.3.2 duration	34
5.10.3.3 lcd	34
5.10.3.4 text	34
6 File Documentation	35
6.1 animLcd.h File Reference	35
6.1.1 Variable Documentation	36
6.1.1.1 LOADING_BAR_OFFSET	36
6.2 animLcd.ino File Reference	36
6.2.1 Detailed Description	37
6.2.2 Variable Documentation	37
6.2.2.1 loading_empty_c	38

6.2.2.2 loading_full_c	38
6.3 animString.h File Reference	38
6.4 animString.ino File Reference	39
6.4.1 Detailed Description	40
6.5 callHandler.h File Reference	40
6.5.1 Detailed Description	41
6.6 callHandler.ino File Reference	41
6.6.1 Detailed Description	42
6.7 customServo.h File Reference	42
6.7.1 Detailed Description	43
6.8 customServo.ino File Reference	43
6.8.1 Detailed Description	43
6.9 header.h File Reference	44
6.9.1 Detailed Description	44
6.9.2 Typedef Documentation	44
6.9.2.1 func_t	44
6.9.2.2 time_t	45
6.10 main.c File Reference	45
6.11 sketch.ino File Reference	45
6.11.1 Detailed Description	47
6.11.2 Macro Definition Documentation	47
6.11.2.1 GEH_ZURUECK	47
6.11.3 Enumeration Type Documentation	47
6.11.3.1 Farbe	47
6.11.4 Function Documentation	48
6.11.4.1 loop()	48
6.11.4.2 mesureColor()	48
6.11.4.3 servosDone()	48
6.11.4.4 setLedColor()	48
6.11.4.5 setup()	48
6.11.4.6 stopButtonClicked()	49
6.11.5 Variable Documentation	49
6.11.5.1 ANGLE_CENTER	49
6.11.5.2 ANGLE_LEFT_HOLE	49
6.11.5.3 ANGLE_MIN	49
6.11.5.4 ANGLE_RIGHT_HOLE	49
6.11.5.5 callHandler	49
6.11.5.6 doFlicker	49
6.11.5.7 lcd	50
6.11.5.8 LOADING_DURATION	50
6.11.5.9 nBlack	50
6.11.5.10 nOrange	50

6.11.5.11 nWhite	50
6.11.5.12 PIN_BLUE	50
6.11.5.13 PIN_GREEN	50
6.11.5.14 PIN_RED	50
6.11.5.15 PIN_SERVO	51
6.11.5.16 PIN_STOPBUTTON	51
6.11.5.17 servo	51
6.11.5.18 SERVO_SPEED_DEFAULT	51
6.11.5.19 SERVO_SPEED_FAST	51
6.11.5.20 stopButton	51
6.12 wokwi-project.txt File Reference	51

Index	53
--------------	-----------

Chapter 1

Main Page

1.1 Einleitung

Hallo, dies ist die Dokumentation für den Code der Sortiermaschine

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ButtonHandler	15
Callable	17
FuncCall	25
LcdString	31
AnimString	13
LcdDotAnim	28
LcdLoadingAnim	30
CallHandler	18
LiquidCrystal_I2C	
AnimatableLcd	9
Servo	
CustomServo	21

Chapter 3

Class Index

3.1 Class List

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

AnimatableLcd	9
AnimString	13
ButtonHandler	15
Callable	17
CallHandler	
Ermöglicht es Calls wie z.B. Funktionen nacheinander aufzurufen, ohne die delay() Funktion zu verwenden	18
CustomServo	
Eine Eigene Servo-Klasse, die es ermöglicht den Servo mit verschiedenen Geschwindigkeiten zu bewegen	21
FuncCall	25
LcdDotAnim	28
LcdLoadingAnim	30
LcdString	31

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

animLcd.h	35
animLcd.ino	36
animString.h	38
animString.ino	39
callHandler.h	
Header datei für den CallHandler	40
callHandler.ino	41
customServo.h	42
customServo.ino	43
header.h	
Definiert variablen-types die überall im Programm benutzt werden	44
main.c	45
sketch.ino	
Hauptdatei	45

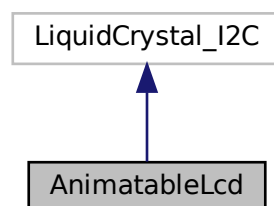
Chapter 5

Class Documentation

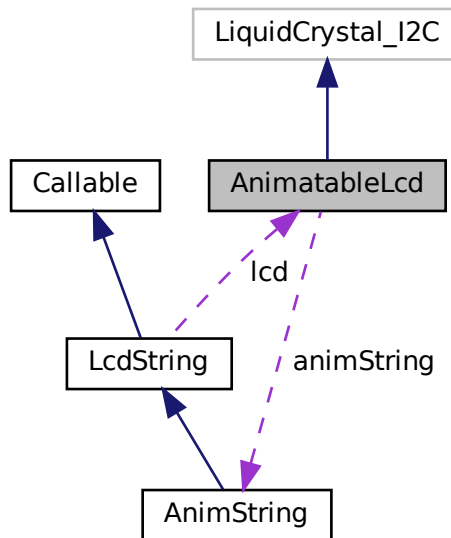
5.1 AnimatableLcd Class Reference

```
#include <animLcd.h>
```

Inheritance diagram for AnimatableLcd:



Collaboration diagram for AnimatableLcd:



Public Member Functions

- void **setAnimation** (**AnimString** * _animString)
setzt die aktuelle animation
- void **printCentered** (String text, int length=-1, int row=0)
Gibt einen String zentriert auf dem Lcd-Display aus.
- void **printPretty** (String text)
gibt den Text "schön" aus, das heißt zentriert und mit automatischen Zeilenumbrüchen
- void **update** ()
wird immer wieder von loop aufgerufen um die Animationen zu updaten
- void **init** ()
Überschreibt die normale lcd init function.
- void **print** (const String &text)
Eigene Lcd-print funktion, die die Möglichkeit bietet eigene Characters in den Text einzufügen.

Public Attributes

- bool **doAnimation** = false

Private Attributes

- **AnimString** * **animString**

5.1.1 Member Function Documentation

5.1.1.1 init()

```
void AnimatableLcd::init ( )
```

Überschreibt die normale lcd init function.

5.1.1.2 print()

```
void AnimatableLcd::print (
    const String & text )
```

Eigene Lcd-print funktion, die die Möglichkeit bietet eigene Characters in den Text einzufügen.

für eigene Character einfach die nummer des Characters in den Text einfügen (**\1n für den nten Character**), \1 für leerzeichen, das nicht in Zeilenumbruch resultiert

Parameters

<i>text</i>	
-------------	--

5.1.1.3 printCentered()

```
void AnimatableLcd::printCentered (
    String text,
    int length = -1,
    int row = 0 )
```

Gibt einen String zentriert auf dem Lcd-Display aus.

Parameters

<i>text</i>	
<i>length</i>	Länge des Textes, wird neu berechnet wenn nicht angegeben
<i>row</i>	Zeile in der der Text ausgegeben werden soll

5.1.1.4 printPretty()

```
void AnimatableLcd::printPretty (
    String text )
```

gibt den Text "schön" aus, das heißt zentriert und mit automatischen Zeilenumbrüchen

Parameters

<i>text</i>	
-------------	--

5.1.1.5 setAnimation()

```
void AnimatableLcd::setAnimation (
    AnimString * _animString )
```

setzt die aktuelle animation

Parameters

<i>_animString</i>	
--------------------	--

5.1.1.6 update()

```
void AnimatableLcd::update ( )
```

wird immer wieder von loop aufgerufen um die Animationen zu updaten

5.1.2 Member Data Documentation

5.1.2.1 animString

```
AnimString* AnimatableLcd::animString [private]
```

5.1.2.2 doAnimation

```
bool AnimatableLcd::doAnimation = false
```

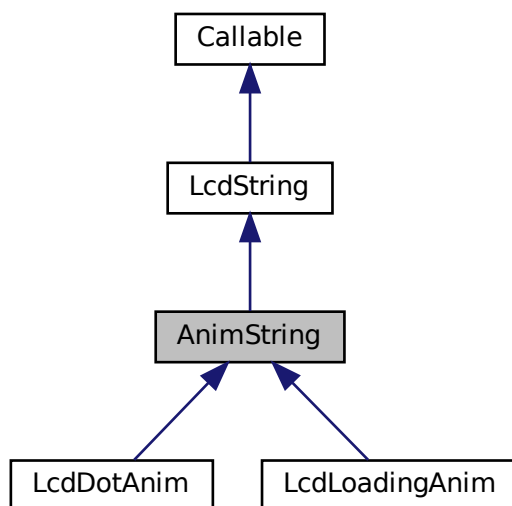
The documentation for this class was generated from the following files:

- [animLcd.h](#)
- [animLcd.ino](#)

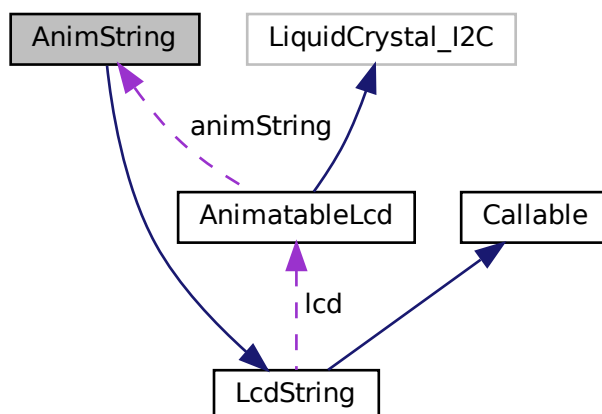
5.2 AnimString Class Reference

```
#include <animString.h>
```

Inheritance diagram for AnimString:



Collaboration diagram for AnimString:



Public Member Functions

- virtual [~AnimString](#) ()
- virtual void [init](#) ()
- void [run](#) ()
 - gibt den String auf dem Lcd-Display aus*
- [LcdString](#) (String [text](#), [AnimatableLcd](#) *[lcd](#), [time_t](#) [duration](#)=0)

Protected Attributes

- [time_t](#) [stepDuration](#)
- [time_t](#) [animStart](#)
- [time_t](#) [lastRefresh](#)

Additional Inherited Members

5.2.1 Constructor & Destructor Documentation

5.2.1.1 ~AnimString()

```
virtual AnimString::~~AnimString ( ) [inline], [virtual]
```

5.2.2 Member Function Documentation

5.2.2.1 init()

```
virtual void AnimString::init ( ) [inline], [virtual]
```

Reimplemented in [LcdDotAnim](#), and [LcdLoadingAnim](#).

5.2.2.2 LcdString()

```
LcdString::LcdString [inline]
```

5.2.2.3 run()

```
void AnimString::run ( ) [virtual]
```

gibt den String auf dem Lcd-Display aus

Reimplemented from [LcdString](#).

5.2.3 Member Data Documentation

5.2.3.1 animStart

```
time_t AnimString::animStart [protected]
```

5.2.3.2 lastRefresh

```
time_t AnimString::lastRefresh [protected]
```

5.2.3.3 stepDuration

```
time_t AnimString::stepDuration [protected]
```

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

- [animString.h](#)
- [animString.ino](#)

5.3 ButtonHandler Class Reference

Public Member Functions

- [ButtonHandler](#) ()
- [ButtonHandler](#) (int [pin](#), void(*[onclick](#))())
- void [update](#) ()

Public Attributes

- void(* [onclick](#))()

Private Attributes

- int `pin`
- bool `isPressed` = false

5.3.1 Constructor & Destructor Documentation

5.3.1.1 ButtonHandler() [1/2]

```
ButtonHandler::ButtonHandler ( ) [inline]
```

5.3.1.2 ButtonHandler() [2/2]

```
ButtonHandler::ButtonHandler (
    int pin,
    void(*)() onclick ) [inline]
```

5.3.2 Member Function Documentation

5.3.2.1 update()

```
void ButtonHandler::update ( ) [inline]
```

5.3.3 Member Data Documentation

5.3.3.1 isPressed

```
bool ButtonHandler::isPressed = false [private]
```

5.3.3.2 onclick

```
void(* ButtonHandler::onclick) ()
```


5.3.3.3 pin

```
int ButtonHandler::pin [private]
```

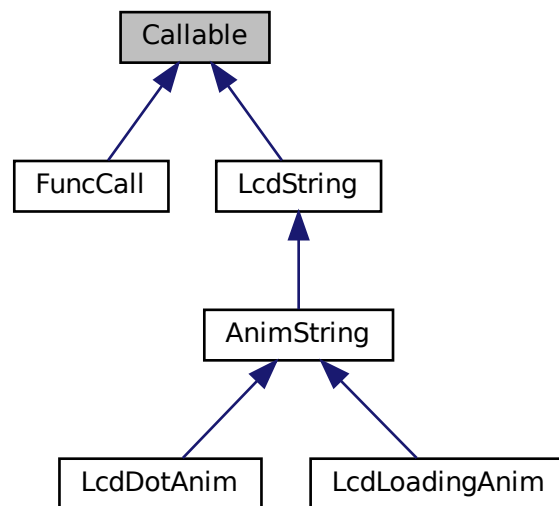
The documentation for this class was generated from the following file:

- [sketch.ino](#)

5.4 Callable Struct Reference

```
#include <animString.h>
```

Inheritance diagram for Callable:



Public Member Functions

- virtual void [run](#) ()
- virtual bool [isDone](#) ()
- virtual [~Callable](#) ()

5.4.1 Constructor & Destructor Documentation

5.4.1.1 ~Callable()

```
virtual Callable::~~Callable ( ) [inline], [virtual]
```

5.4.2 Member Function Documentation

5.4.2.1 isDone()

```
virtual bool Callable::isDone ( ) [inline], [virtual]
```

Reimplemented in [LcdString](#), and [FuncCall](#).

5.4.2.2 run()

```
virtual void Callable::run ( ) [inline], [virtual]
```

Reimplemented in [AnimString](#), [LcdString](#), and [FuncCall](#).

The documentation for this struct was generated from the following file:

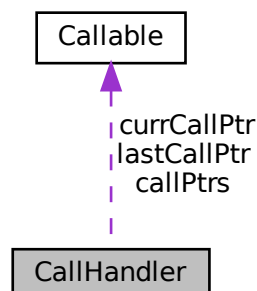
- [animString.h](#)

5.5 CallHandler Class Reference

Ermöglicht es Calls wie z.B. Funktionen nacheinander aufzurufen, ohne die delay() Funktion zu verwenden.

```
#include <callHandler.h>
```

Collaboration diagram for CallHandler:



Public Member Functions

- void `deleteCalls` ()
setzt den Speicherplatz der von den Calls besetzt wurde frei
- void `setCalls` (Callable *newCallPtrs[], size_t nCalls)
Setzt die neuen Calls, die ausgeführt werden sollen.
- virtual `~ CallHandler` ()
- void `update` ()
wird von loop aufgerufen, ermöglicht es zu überprüfen, wechselt zum nächsten Call, wenn der Aktuelle vorbei ist

Public Attributes

- bool `running` = false

Private Attributes

- Callable ** `callPtrs`
- Callable ** `currCallPtr`
- Callable ** `lastCallPtr`
- time_t `lastCallT`
- bool `callsSet` = false

5.5.1 Detailed Description

Ermöglicht es Calls wie z.B. Funktionen nacheinander aufzurufen, ohne die `delay()` Funktion zu verwenden.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 `~ CallHandler()`

```
virtual CallHandler::~~ CallHandler ( ) [virtual]
```

5.5.3 Member Function Documentation

5.5.3.1 `deleteCalls()`

```
void CallHandler::deleteCalls ( )
```

setzt den Speicherplatz der von den Calls besetzt wurde frei

5.5.3.2 `setCalls()`

```
void CallHandler::setCalls (
    Callable * newCallPtrs[],
    size_t nCalls )
```

Setzt die neuen Calls, die ausgeführt werden sollen.

Parameters

<i>newCallPtrs</i>	
<i>nCalls</i>	

5.5.3.3 update()

```
void CallHandler::update ( )
```

wird von loop aufgerufen, ermöglicht es zu überprüfen, wechselt zum nächsten Call, wenn der Aktuelle vorbei ist

5.5.4 Member Data Documentation**5.5.4.1 callPtrs**

```
Callable** CallHandler::callPtrs [private]
```

5.5.4.2 callsSet

```
bool CallHandler::callsSet = false [private]
```

5.5.4.3 currCallPtr

```
Callable** CallHandler::currCallPtr [private]
```

5.5.4.4 lastCallPtr

```
Callable** CallHandler::lastCallPtr [private]
```

5.5.4.5 lastCallT

```
time_t CallHandler::lastCallT [private]
```

5.5.4.6 running

```
bool CallHandler::running = false
```

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

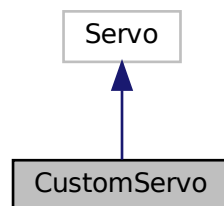
- [callHandler.h](#)
- [callHandler.ino](#)

5.6 CustomServo Class Reference

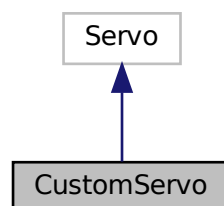
Eine Eigene Servo-Klasse, die es ermöglicht den Servo mit verschiedenen Geschwindigkeiten zu bewegen.

```
#include <customServo.h>
```

Inheritance diagram for CustomServo:



Collaboration diagram for CustomServo:



Public Member Functions

- void `write` (short newAngle)
bewegt den Servo mit einer vorher spezifizierten Geschwindigkeit
- void `write` (short newAngle, `time_t` duration)
*bewegt den Servo in **duration** ms*
- void `writeDirect` (short angle)
- void `setSpeed` (float newSpeed)
- void `updatePos` ()
- void `stop` ()
stoppt den Servo
- void `start` ()
lässt den Servo weiterlaufen
- bool `isDone` ()
Gibt an, ob der Servo angekommen ist.

Public Attributes

- bool `done`

Private Member Functions

- void `startMove` ()
Setzt Variablen, die benötigt werden um den Servo zu bewegen.

Private Attributes

- short `startAngle`
- short `targetAngle`
- float `speed`
- `time_t` `startTime`
- Servo `servo`

5.6.1 Detailed Description

Eine Eigene Servo-Klasse, die es ermöglicht den Servo mit verschiedenen Geschwindigkeiten zu bewegen.

5.6.2 Member Function Documentation

5.6.2.1 isDone()

```
bool CustomServo::isDone ( )
```

Gibt an, ob der Servo angekommen ist.

Returns

true
false

5.6.2.2 setSpeed()

```
void CustomServo::setSpeed (
    float newSpeed )
```

5.6.2.3 start()

```
void CustomServo::start ( )
```

lässt den Servo weiterlaufen

5.6.2.4 startMove()

```
void CustomServo::startMove ( ) [private]
```

Setzt Variablen, die benötigt werden um den Servo zu bewegen.

5.6.2.5 stop()

```
void CustomServo::stop ( )
```

stoppt den Servo

5.6.2.6 updatePos()

```
void CustomServo::updatePos ( )
```

5.6.2.7 write() [1/2]

```
void CustomServo::write (
    short newAngle )
```

bewegt den Servo mit einer vorher spezifizierten Geschwindigkeit

Parameters

<i>newAngle</i>	
-----------------	--

5.6.2.8 write() [2/2]

```
void CustomServo::write (  
    short newAngle,  
    time_t duration )
```

bewegt den Servo in **duration** ms

Parameters

<i>newAngle</i>	
<i>duration</i>	

5.6.2.9 writeDirect()

```
void CustomServo::writeDirect (  
    short angle )
```

5.6.3 Member Data Documentation**5.6.3.1 done**

```
bool CustomServo::done
```

5.6.3.2 servo

```
Servo CustomServo::servo [private]
```

5.6.3.3 speed

```
float CustomServo::speed [private]
```


5.6.3.4 startAngle

```
short CustomServo::startAngle [private]
```

5.6.3.5 startTime

```
time_t CustomServo::startTime [private]
```

5.6.3.6 targetAngle

```
short CustomServo::targetAngle [private]
```

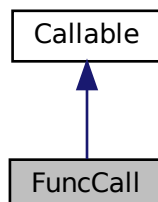
The documentation for this class was generated from the following files:

- [customServo.h](#)
- [customServo.ino](#)

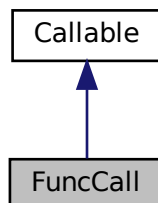
5.7 FuncCall Struct Reference

```
#include <animString.h>
```

Inheritance diagram for FuncCall:



Collaboration diagram for FuncCall:



Public Member Functions

- [FuncCall](#) ([func_t](#)< void > [call](#), [func_t](#)< bool > [_isDone](#))
- [FuncCall](#) ([func_t](#)< void > [call](#))
- virtual [~FuncCall](#) ()
- void [run](#) ()
ruft die angegebene Funktion auf
- bool [isDone](#) ()
Gibt zurück, ob der nächste Call ausgeführt werden sollte.

Public Attributes

- [func_t](#)< void > [call](#)
- [func_t](#)< bool > [_isDone](#)

5.7.1 Constructor & Destructor Documentation

5.7.1.1 FuncCall() [1/2]

```
FuncCall::FuncCall (  
    func\_t< void > call,  
    func\_t< bool > \_isDone ) [inline]
```

5.7.1.2 FuncCall() [2/2]

```
FuncCall::FuncCall (  
    func\_t< void > call ) [inline]
```

5.7.1.3 ~FuncCall()

```
virtual FuncCall::~~FuncCall ( ) [inline], [virtual]
```

5.7.2 Member Function Documentation

5.7.2.1 isDone()

```
bool FuncCall::isDone ( ) [virtual]
```

Gibt zurück, ob der nächste Call ausgeführt werden sollte.

Returns

true
false

Reimplemented from [Callable](#).

5.7.2.2 run()

```
void FuncCall::run ( ) [virtual]
```

ruft die angegebene Funktion auf

Reimplemented from [Callable](#).

5.7.3 Member Data Documentation

5.7.3.1 _isDone

```
func_t<bool> FuncCall::_isDone
```

5.7.3.2 call

```
func_t<void> FuncCall::call
```

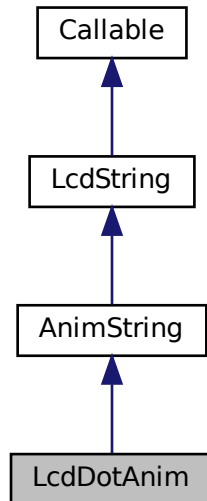
The documentation for this struct was generated from the following files:

- [animString.h](#)
- [animString.ino](#)

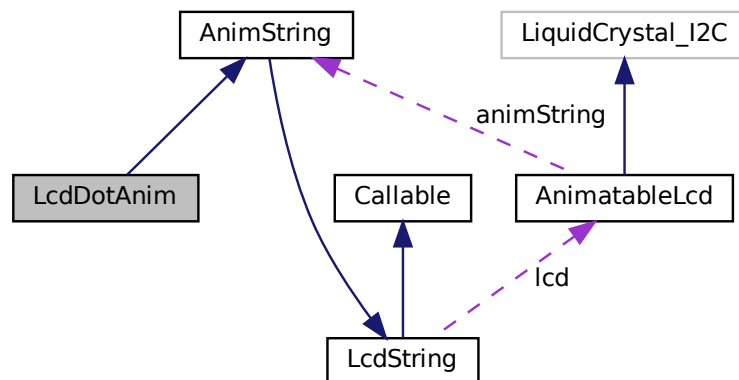
5.8 LcdDotAnim Class Reference

```
#include <animString.h>
```

Inheritance diagram for LcdDotAnim:



Collaboration diagram for LcdDotAnim:



Public Member Functions

- **LcdDotAnim** (String **text**, **AnimatableLcd** ***lcd**, **time_t** **duration**=0, **time_t** **_stepDuration**=500)
- void **init** ()
- void **update** ()

Additional Inherited Members

5.8.1 Constructor & Destructor Documentation

5.8.1.1 LcdDotAnim()

```
LcdDotAnim::LcdDotAnim (
    String text,
    AnimatableLcd * lcd,
    time_t duration = 0,
    time_t _stepDuration = 500 ) [inline]
```

5.8.2 Member Function Documentation

5.8.2.1 init()

```
void LcdDotAnim::init ( ) [virtual]
```

Reimplemented from [AnimString](#).

5.8.2.2 update()

```
void LcdDotAnim::update ( ) [virtual]
```

Reimplemented from [LcdString](#).

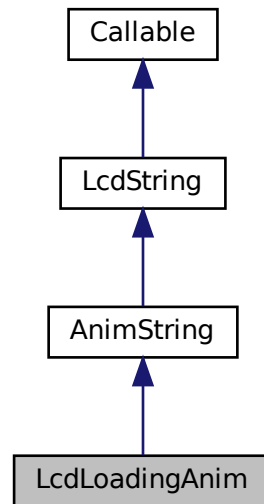
The documentation for this class was generated from the following files:

- [animString.h](#)
- [animString.ino](#)

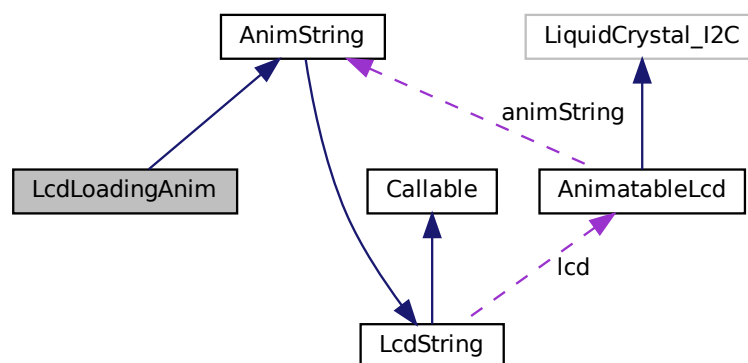
5.9 LcdLoadingAnim Class Reference

```
#include <animString.h>
```

Inheritance diagram for LcdLoadingAnim:



Collaboration diagram for LcdLoadingAnim:



Public Member Functions

- void **init** ()
- void **update** ()

Additional Inherited Members

5.9.1 Member Function Documentation

5.9.1.1 init()

```
void LcdLoadingAnim::init ( ) [virtual]
```

Reimplemented from [AnimString](#).

5.9.1.2 update()

```
void LcdLoadingAnim::update ( ) [virtual]
```

Reimplemented from [LcdString](#).

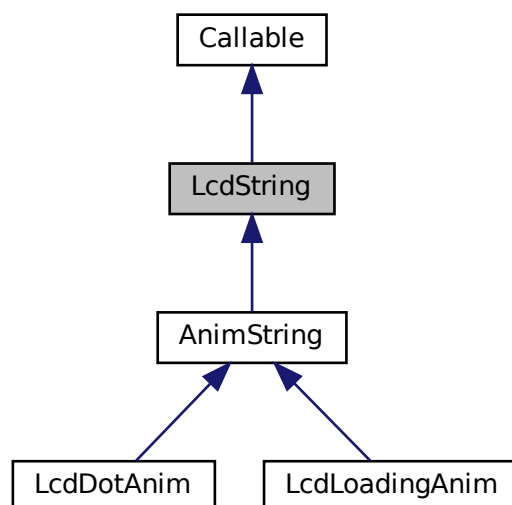
The documentation for this class was generated from the following files:

- [animString.h](#)
- [animString.ino](#)

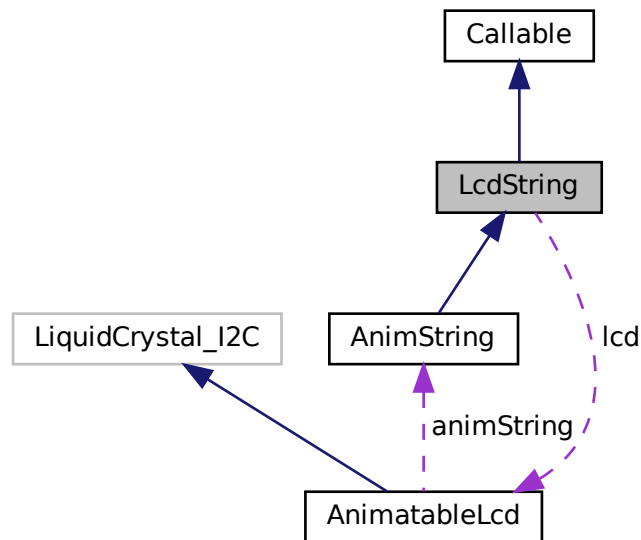
5.10 LcdString Struct Reference

```
#include <animString.h>
```

Inheritance diagram for LcdString:



Collaboration diagram for LcdString:



Public Member Functions

- **LcdString** (String **text**, AnimatableLcd ***lcd**, time_t **duration**=0)
- virtual ~**LcdString** ()
- bool **isDone** ()
Gibt zurück, ob die duration überschritten ist.
- virtual void **run** ()
gibt den String auf dem Lcd-Display aus
- virtual void **update** ()

Public Attributes

- String **text**
- AnimatableLcd * **lcd**
- time_t **duration**
- time_t **callStart**

5.10.1 Constructor & Destructor Documentation

5.10.1.1 LcdString()

```
LcdString::LcdString (
    String text,
    AnimatableLcd * lcd,
    time_t duration = 0 ) [inline]
```

5.10.1.2 ~LcdString()

```
virtual LcdString::~~LcdString ( ) [inline], [virtual]
```

5.10.2 Member Function Documentation

5.10.2.1 isDone()

```
bool LcdString::isDone ( ) [virtual]
```

Gibt zurück, ob die duration überschritten ist.

Returns

true
false

Reimplemented from [Callable](#).

5.10.2.2 run()

```
void LcdString::run ( ) [virtual]
```

gibt den String auf dem Lcd-Display aus

Reimplemented from [Callable](#).

Reimplemented in [AnimString](#).

5.10.2.3 update()

```
virtual void LcdString::update ( ) [inline], [virtual]
```

Reimplemented in [LcdDotAnim](#), and [LcdLoadingAnim](#).

5.10.3 Member Data Documentation

5.10.3.1 callStart

`time_t` LcdString::callStart

5.10.3.2 duration

`time_t` LcdString::duration

5.10.3.3 lcd

`AnimatableLcd*` LcdString::lcd

5.10.3.4 text

`String` LcdString::text

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

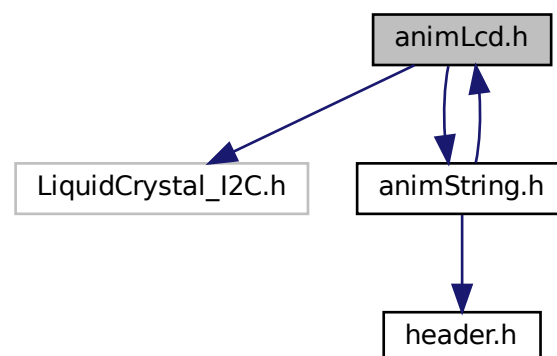
- [animString.h](#)
- [animString.ino](#)

Chapter 6

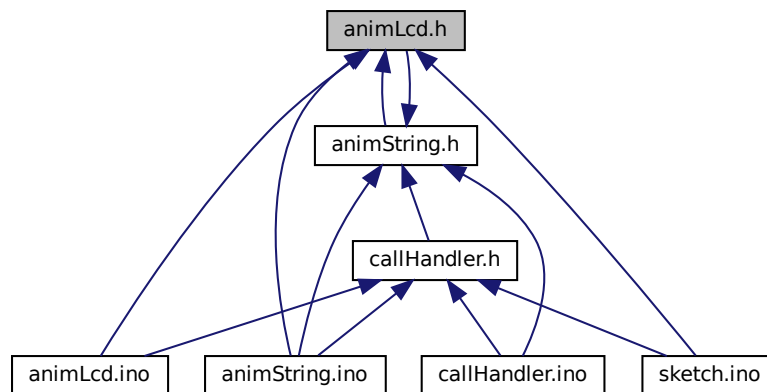
File Documentation

6.1 animLcd.h File Reference

```
#include <LiquidCrystal_I2C.h>  
#include "animString.h"  
Include dependency graph for animLcd.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [AnimatableLcd](#)

Variables

- const int [LOADING_BAR_OFFSET](#) = 2

6.1.1 Variable Documentation

6.1.1.1 LOADING_BAR_OFFSET

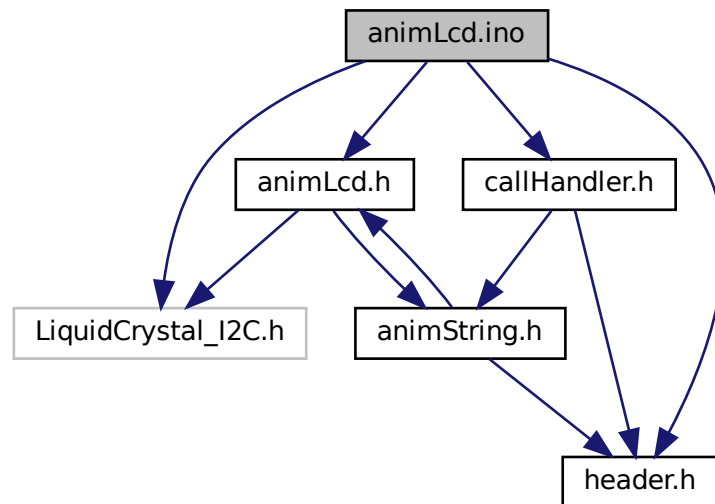
```
const int LOADING_BAR_OFFSET = 2
```

6.2 animLcd.ino File Reference

```
#include <LiquidCrystal_I2C.h>
#include "header.h"
#include "animLcd.h"
```

```
#include "callHandler.h"
```

Include dependency graph for animLcd.ino:



Variables

- const byte `loading_empty_c` [8]
- const byte `loading_full_c` [8]

6.2.1 Detailed Description

Author

Arne de Borman

Version

0.1

Date

2022-05-26

6.2.2 Variable Documentation

6.2.2.1 loading_empty_c

```
const byte loading_empty_c[8]
```

Initial value:

```
= {  
    B11111,  
    B10001,  
    B10001,  
    B10001,  
    B10001,  
    B10001,  
    B10001,  
    B10001,  
    B11111  
}
```

6.2.2.2 loading_full_c

```
const byte loading_full_c[8]
```

Initial value:

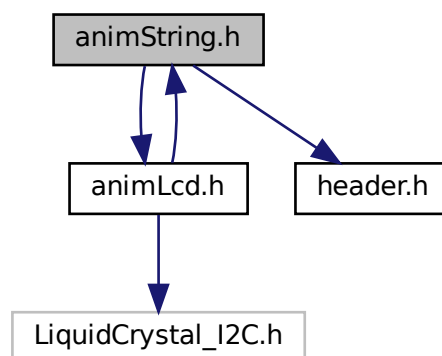
```
= {  
    B11111,  
    B11111,  
    B11111,  
    B11111,  
    B11111,  
    B11111,  
    B11111,  
    B11111  
}
```

6.3 animString.h File Reference

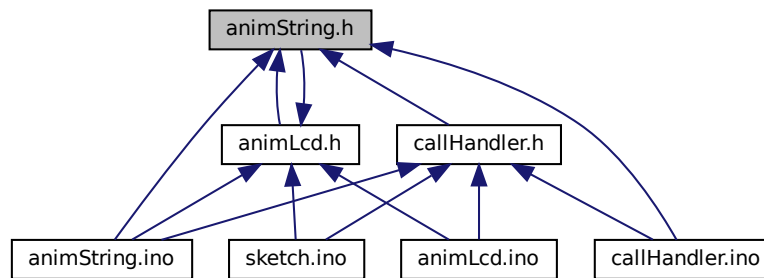
```
#include "animLcd.h"
```

```
#include "header.h"
```

Include dependency graph for animString.h:



This graph shows which files directly or indirectly include this file:



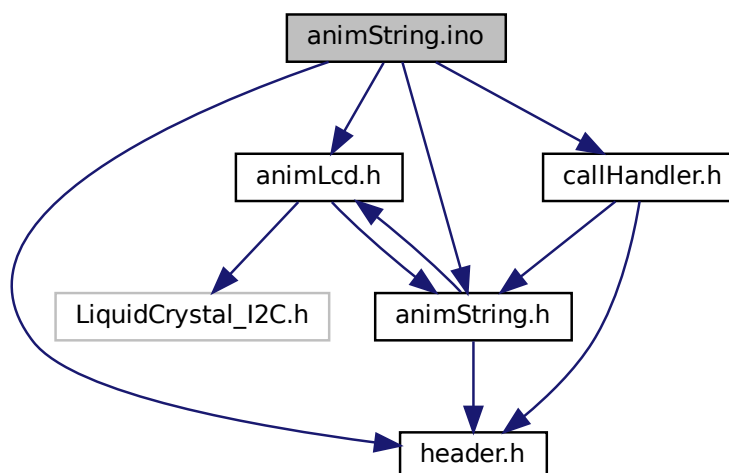
Classes

- struct [Callable](#)
- struct [FuncCall](#)
- struct [LcdString](#)
- class [AnimString](#)
- class [LcdLoadingAnim](#)
- class [LcdDotAnim](#)

6.4 animString.ino File Reference

```
#include "animLcd.h"
#include "callHandler.h"
#include "header.h"
#include "animString.h"
```

Include dependency graph for animString.ino:



6.4.1 Detailed Description

Author

Arne de Borman

Version

0.1

Date

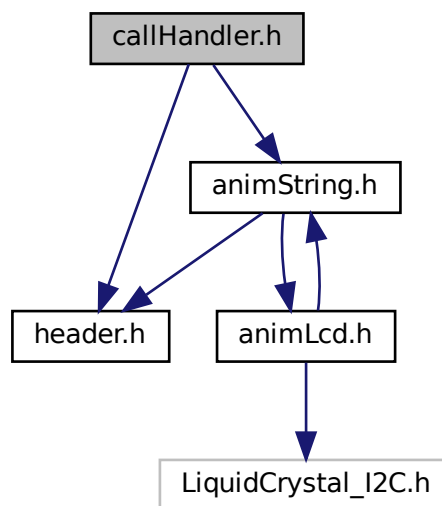
2022-05-26

6.5 callHandler.h File Reference

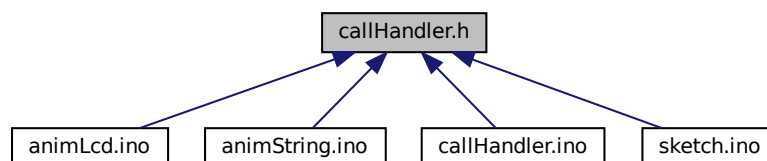
header datei für den [CallHandler](#)

```
#include "header.h"  
#include "animString.h"
```

Include dependency graph for callHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CallHandler](#)

Ermöglicht es Calls wie z.B. Funktionen nacheinander aufzurufen, ohne die delay() Funktion zu verwenden.

6.5.1 Detailed Description

header datei für den [CallHandler](#)

Author

your name (you@domain.com)

Version

0.1

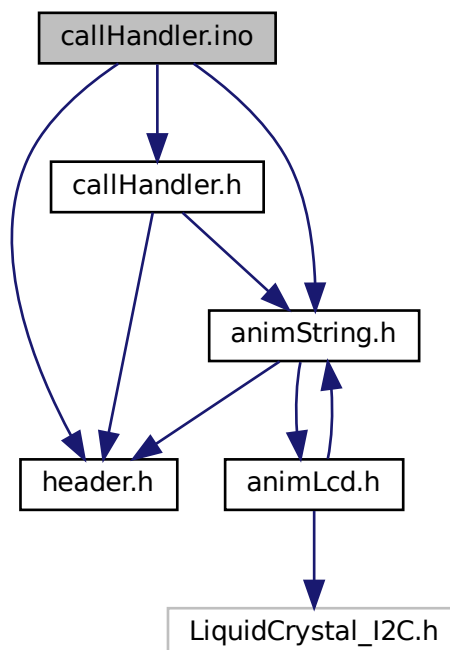
Date

2022-05-26

6.6 callHandler.ino File Reference

```
#include "header.h"
#include "animString.h"
#include "callHandler.h"
```

Include dependency graph for callHandler.ino:



6.6.1 Detailed Description

Author

Arne de Borman

Version

0.1

Date

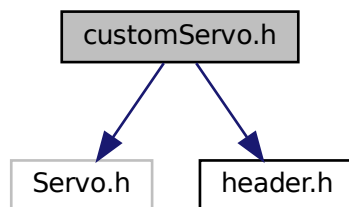
2022-05-26

6.7 customServo.h File Reference

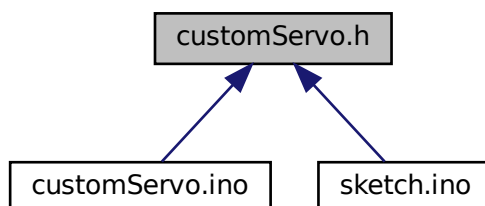
```
#include <Servo.h>
```

```
#include "header.h"
```

Include dependency graph for customServo.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CustomServo](#)

Eine Eigene Servo-Klasse, die es ermöglicht den Servo mit verschiedenen Geschwindigkeiten zu bewegen.

6.7.1 Detailed Description

Version

0.1

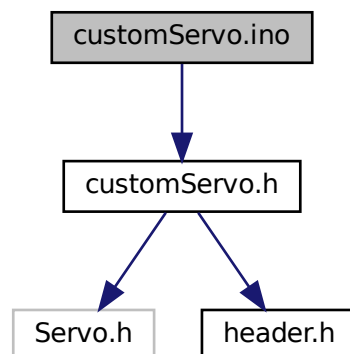
Date

2022-05-26

6.8 customServo.ino File Reference

```
#include "customServo.h"
```

Include dependency graph for customServo.ino:



6.8.1 Detailed Description

Author

Arne de borman

Version

0.1

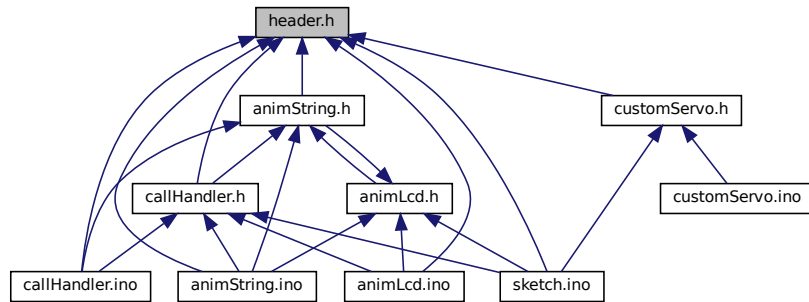
Date

2022-05-26

6.9 header.h File Reference

definiert variablen-types die überall im Programm benutzt werden

This graph shows which files directly or indirectly include this file:



Typedefs

- using `time_t` = unsigned long
Ein Zeit typ.
- template<typename ReturnT = void>
using `func_t` = ReturnT(*)()
ein Funktions-typ

6.9.1 Detailed Description

definiert variablen-types die überall im Programm benutzt werden

Version

0.1

Date

2022-05-26

6.9.2 Typedef Documentation

6.9.2.1 func_t

```
template<typename ReturnT = void>
using func_t = ReturnT(*)()
```

ein Funktions-typ

Template Parameters

<i>ReturnT</i>	der Rückgabewert der Funktion die referenziert wird
----------------	---

6.9.2.2 time_t

```
using time_t = unsigned long
```

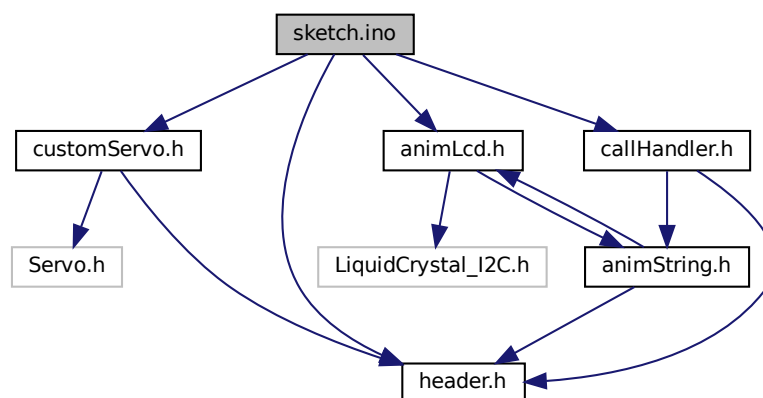
Ein Zeit typ.

6.10 main.c File Reference

6.11 sketch.ino File Reference

Hauptdatei.

```
#include "customServo.h"
#include "animLcd.h"
#include "callHandler.h"
#include "header.h"
Include dependency graph for sketch.ino:
```



Classes

- class [ButtonHandler](#)

Macros

- `#define GEH_ZURUECK`

Enumerations

- `enum Farbe { WHITE, BLACK, ORANGE, NOTHING }`

Functions

- `Farbe measureColor ()`
- `void setLedColor (unsigned char r, unsigned char g, unsigned char b)`
- `void stopButtonClicked ()`
wird ausgeführt wenn der Stop-Knopf gedrückt wird
- `bool servosDone ()`
hilf Function, Methoden können nicht als functionsparameter benutzt werden
- `void setup ()`
wird am Anfang des Programms aufgerufen
- `void loop ()`
wird immer wieder ausgeführt

Variables

- `const int LOADING_DURATION = 3000`
- `const int ANGLE_LEFT_HOLE = 180`
- `const int ANGLE_RIGHT_HOLE = 90`
- `const int ANGLE_CENTER = 130`
- `const int ANGLE_MIN = 45`
- `const int PIN_SERVO = 6`
- `const int PIN_STOPBUTTON = 13`
- `const int PIN_RED = 11`
- `const int PIN_GREEN = 10`
- `const int PIN_BLUE = 9`
- `const float SERVO_SPEED_DEFAULT = 0.01f`
- `const float SERVO_SPEED_FAST = 0.5f`
- `AnimatableLcd lcd (0x27, 16, 2)`
- `CallHandler callHandler`
- `CustomServo servo`
- `int nWhite = 0`
- `int nBlack = 0`
- `int nOrange = 0`
- `bool doFlicker = false`
- `ButtonHandler stopButton`
der Stop Knopf

6.11.1 Detailed Description

Hauptdatei.

Author

Arne de Borman

Version

0.1

Date

2022-05-26

6.11.2 Macro Definition Documentation

6.11.2.1 GEH_ZURUECK

```
#define GEH_ZURUECK
```

Value:

```
new LcdDotAnim("Gehe zur\365ck",&lcd),\
new FuncCall([]() {\
    servo.setSpeed(SERVO_SPEED_FAST);\
    servo.write(ANGLE_CENTER);\
}, &servoIsDone)
```

6.11.3 Enumeration Type Documentation

6.11.3.1 Farbe

```
enum Farbe
```

Enumerator

WHITE	
BLACK	
ORANGE	
NOTHING	

6.11.4 Function Documentation

6.11.4.1 loop()

```
void loop ( )
```

wird immer wieder ausgeführt

6.11.4.2 measureColor()

```
Farbe measureColor ( )
```

6.11.4.3 servosDone()

```
bool servoIsDone ( )
```

hilf Function, Methoden können nicht als functionsparameter benutzt werden

Returns

true

false

6.11.4.4 setLedColor()

```
void setLedColor (
    unsigned char r,
    unsigned char g,
    unsigned char b )
```

6.11.4.5 setup()

```
void setup ( )
```

wird am Anfang des Programms aufgerufen

6.11.4.6 stopButtonClicked()

```
void stopButtonClicked ( )
```

wird ausgeführt wenn der Stop-Knopf gedrückt wird

6.11.5 Variable Documentation

6.11.5.1 ANGLE_CENTER

```
const int ANGLE_CENTER = 130
```

6.11.5.2 ANGLE_LEFT_HOLE

```
const int ANGLE_LEFT_HOLE = 180
```

6.11.5.3 ANGLE_MIN

```
const int ANGLE_MIN = 45
```

6.11.5.4 ANGLE_RIGHT_HOLE

```
const int ANGLE_RIGHT_HOLE = 90
```

6.11.5.5 callHandler

```
CallHandler callHandler
```

6.11.5.6 doFlicker

```
bool doFlicker =false
```

6.11.5.7 lcd

```
AnimatableLcd lcd(0x27, 16, 2)
```

6.11.5.8 LOADING_DURATION

```
const int LOADING_DURATION = 3000
```

6.11.5.9 nBlack

```
int nBlack = 0
```

6.11.5.10 nOrange

```
int nOrange = 0
```

6.11.5.11 nWhite

```
int nWhite = 0
```

6.11.5.12 PIN_BLUE

```
const int PIN_BLUE = 9
```

6.11.5.13 PIN_GREEN

```
const int PIN_GREEN = 10
```

6.11.5.14 PIN_RED

```
const int PIN_RED = 11
```

6.11.5.15 PIN_SERVO

```
const int PIN_SERVO = 6
```

6.11.5.16 PIN_STOPBUTTON

```
const int PIN_STOPBUTTON = 13
```

6.11.5.17 servo

```
CustomServo servo
```

6.11.5.18 SERVO_SPEED_DEFAULT

```
const float SERVO_SPEED_DEFAULT = 0.01f
```

6.11.5.19 SERVO_SPEED_FAST

```
const float SERVO_SPEED_FAST = 0.5f
```

6.11.5.20 stopButton

```
ButtonHandler stopButton
```

der Stop Knopf

6.12 wokwi-project.txt File Reference

Index

- _isDone
 - FuncCall, [27](#)
 - ~ CallHandler
 - CallHandler, [19](#)
 - ~AnimString
 - AnimString, [14](#)
 - ~Callable
 - Callable, [17](#)
 - ~FuncCall
 - FuncCall, [26](#)
 - ~LcdString
 - LcdString, [33](#)
- ANGLE_CENTER
 - sketch.ino, [49](#)
- ANGLE_LEFT_HOLE
 - sketch.ino, [49](#)
- ANGLE_MIN
 - sketch.ino, [49](#)
- ANGLE_RIGHT_HOLE
 - sketch.ino, [49](#)
- AnimatableLcd, [9](#)
 - animString, [12](#)
 - doAnimation, [12](#)
 - init, [11](#)
 - print, [11](#)
 - printCentered, [11](#)
 - printPretty, [11](#)
 - setAnimation, [12](#)
 - update, [12](#)
- animLcd.h, [35](#)
 - LOADING_BAR_OFFSET, [36](#)
- animLcd.ino, [36](#)
 - loading_empty_c, [37](#)
 - loading_full_c, [38](#)
- animStart
 - AnimString, [15](#)
- AnimString, [13](#)
 - ~AnimString, [14](#)
 - animStart, [15](#)
 - init, [14](#)
 - lastRefresh, [15](#)
 - LcdString, [14](#)
 - run, [14](#)
 - stepDuration, [15](#)
- animString
 - AnimatableLcd, [12](#)
- animString.h, [38](#)
- animString.ino, [39](#)
- BLACK
 - sketch.ino, [47](#)
- ButtonHandler, [15](#)
 - ButtonHandler, [16](#)
 - isPressed, [16](#)
 - onclick, [16](#)
 - pin, [16](#)
 - update, [16](#)
- call
 - FuncCall, [27](#)
- Callable, [17](#)
 - ~Callable, [17](#)
 - isDone, [18](#)
 - run, [18](#)
- CallHandler, [18](#)
 - ~ CallHandler, [19](#)
 - callPtrs, [20](#)
 - callsSet, [20](#)
 - currCallPtr, [20](#)
 - deleteCalls, [19](#)
 - lastCallPtr, [20](#)
 - lastCallT, [20](#)
 - running, [20](#)
 - setCalls, [19](#)
 - update, [20](#)
- callHandler
 - sketch.ino, [49](#)
- callHandler.h, [40](#)
- callHandler.ino, [41](#)
- callPtrs
 - CallHandler, [20](#)
- callsSet
 - CallHandler, [20](#)
- callStart
 - LcdString, [34](#)
- currCallPtr
 - CallHandler, [20](#)
- CustomServo, [21](#)
 - done, [24](#)
 - isDone, [22](#)
 - servo, [24](#)
 - setSpeed, [23](#)
 - speed, [24](#)
 - start, [23](#)
 - startAngle, [24](#)
 - startMove, [23](#)
 - startTime, [25](#)
 - stop, [23](#)
 - targetAngle, [25](#)

- updatePos, 23
 - write, 23, 24
 - writeDirect, 24
- customServo.h, 42
- customServo.ino, 43
- deleteCalls
 - CallHandler, 19
- doAnimation
 - AnimatableLcd, 12
- doFlicker
 - sketch.ino, 49
- done
 - CustomServo, 24
- duration
 - LcdString, 34
- Farbe
 - sketch.ino, 47
- func_t
 - header.h, 44
- FuncCall, 25
 - _isDone, 27
 - ~FuncCall, 26
 - call, 27
 - FuncCall, 26
 - isDone, 26
 - run, 27
- GEH_ZURUECK
 - sketch.ino, 47
- header.h, 44
 - func_t, 44
 - time_t, 45
- init
 - AnimatableLcd, 11
 - AnimString, 14
 - LcdDotAnim, 29
 - LcdLoadingAnim, 31
- isDone
 - Callable, 18
 - CustomServo, 22
 - FuncCall, 26
 - LcdString, 33
- isPressed
 - ButtonHandler, 16
- lastCallPtr
 - CallHandler, 20
- lastCallT
 - CallHandler, 20
- lastRefresh
 - AnimString, 15
- lcd
 - LcdString, 34
 - sketch.ino, 49
- LcdDotAnim, 28
 - init, 29
- LcdDotAnim, 29
 - update, 29
- LcdLoadingAnim, 30
 - init, 31
 - update, 31
- LcdString, 31
 - ~LcdString, 33
 - AnimString, 14
 - callStart, 34
 - duration, 34
 - isDone, 33
 - lcd, 34
 - LcdString, 32
 - run, 33
 - text, 34
 - update, 33
- LOADING_BAR_OFFSET
 - animLcd.h, 36
- LOADING_DURATION
 - sketch.ino, 50
- loading_empty_c
 - animLcd.ino, 37
- loading_full_c
 - animLcd.ino, 38
- loop
 - sketch.ino, 48
- main.c, 45
- measureColor
 - sketch.ino, 48
- nBlack
 - sketch.ino, 50
- nOrange
 - sketch.ino, 50
- NOTHING
 - sketch.ino, 47
- nWhite
 - sketch.ino, 50
- onclick
 - ButtonHandler, 16
- ORANGE
 - sketch.ino, 47
- pin
 - ButtonHandler, 16
- PIN_BLUE
 - sketch.ino, 50
- PIN_GREEN
 - sketch.ino, 50
- PIN_RED
 - sketch.ino, 50
- PIN_SERVO
 - sketch.ino, 50
- PIN_STOPBUTTON
 - sketch.ino, 51
- print
 - AnimatableLcd, 11

- printCentered
 - AnimatableLcd, 11
- printPretty
 - AnimatableLcd, 11
- run
 - AnimString, 14
 - Callable, 18
 - FuncCall, 27
 - LcdString, 33
- running
 - CallHandler, 20
- servo
 - CustomServo, 24
 - sketch.ino, 51
- SERVO_SPEED_DEFAULT
 - sketch.ino, 51
- SERVO_SPEED_FAST
 - sketch.ino, 51
- servosDone
 - sketch.ino, 48
- setAnimation
 - AnimatableLcd, 12
- setCalls
 - CallHandler, 19
- setLedColor
 - sketch.ino, 48
- setSpeed
 - CustomServo, 23
- setup
 - sketch.ino, 48
- sketch.ino, 45
 - ANGLE_CENTER, 49
 - ANGLE_LEFT_HOLE, 49
 - ANGLE_MIN, 49
 - ANGLE_RIGHT_HOLE, 49
 - BLACK, 47
 - callHandler, 49
 - doFlicker, 49
 - Farbe, 47
 - GEH_ZURUECK, 47
 - lcd, 49
 - LOADING_DURATION, 50
 - loop, 48
 - measureColor, 48
 - nBlack, 50
 - nOrange, 50
 - NOTHING, 47
 - nWhite, 50
 - ORANGE, 47
 - PIN_BLUE, 50
 - PIN_GREEN, 50
 - PIN_RED, 50
 - PIN_SERVO, 50
 - PIN_STOPBUTTON, 51
 - servo, 51
 - SERVO_SPEED_DEFAULT, 51
 - SERVO_SPEED_FAST, 51
 - servosDone, 48
 - setLedColor, 48
 - setup, 48
 - stopButton, 51
 - stopButtonClicked, 48
 - WHITE, 47
- speed
 - CustomServo, 24
- start
 - CustomServo, 23
- startAngle
 - CustomServo, 24
- startMove
 - CustomServo, 23
- startTime
 - CustomServo, 25
- stepDuration
 - AnimString, 15
- stop
 - CustomServo, 23
- stopButton
 - sketch.ino, 51
- stopButtonClicked
 - sketch.ino, 48
- targetAngle
 - CustomServo, 25
- text
 - LcdString, 34
- time_t
 - header.h, 45
- update
 - AnimatableLcd, 12
 - ButtonHandler, 16
 - CallHandler, 20
 - LcdDotAnim, 29
 - LcdLoadingAnim, 31
 - LcdString, 33
- updatePos
 - CustomServo, 23
- WHITE
 - sketch.ino, 47
- wokwi-project.txt, 51
- write
 - CustomServo, 23, 24
- writeDirect
 - CustomServo, 24