Library to Control a

shiftregister 74HC595

Lk-16
Version 1.0
Sun Sep 4 2022

Class Index

Class List

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

File Index

File List

Here is a list of all files with brief descriptions:

Class Documentation

shiftregister Class Reference

```
#include <shiftregister.h>
```

Public Member Functions

- **shiftregister** (int numberOfRegisters, int sh, int st, int ds)
- void **setPin** (int pinNo, boolean pinStatus)
- void **printPinStatus** ()
- boolean **getPinStatus** (int pinNo)

Constructor & Destructor Documentation

shiftregister::shiftregister (int *numberOfRegisters*, int *sh*, int *st*, int *ds*)

Member Function Documentation

boolean shiftregister::getPinStatus (int *pinNo*)

void shiftregister::printPinStatus ()

void shiftregister::setPin (int *pinNo*, boolean *pinStatus*)

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

- shiftregister/src/**shiftregister.h**
- shiftregister/src/**shiftregister.cpp**

shiftregister.h

Go to the documentation of this file.

```
1 /*
2  *  @name shiftregister
3  *  @author lk-16
4  *  @date 14.10.2020
5  *  This libary allows to control a shift register easily.
6  *
7 */
8
9
10
11 #ifndef shiftregister_h
12 #define shiftregister_h
13 #include "Arduino.h"
14
15 class shiftregister
16 {
17   public:
18
19     shiftregister(int numberOfRegisters, int sh, int st, int ds);
20     void setPin(int pinNo, boolean pinStatus);              /*<Pins can be controlled
individually*/
21     void printPinStatus();                                   /*<Status of the entire
pins is printed via the serial monitor*/
22     boolean getPinStatus(int pinNo);                         /*<Returns the status of
one pin*/
23
24   private:
25     int _sh;
26     int _st;
27     int _ds;
28     int _numberOfPins;                                        //number of Pins in the
register
29     boolean *_data;                                         //flexible array for
passing the data to the register
30
31
32 };
33
34 #endif
```