

A500KB

Generated by Doxygen 1.8.9.1

Tue Sep 22 2015 11:44:40

Contents

1	Main Page	1
2	LICENSE	1
3	Class Index	1
3.1	Class List	2
4	Class Documentation	2
4.1	A500KB Class Reference	2
4.1.1	Constructor & Destructor Documentation	2
4.1.2	Member Function Documentation	2
Index		5

1 Main Page

Attach an Amiga A500 (or similar from other models) keyboard to an Arduino or chipKIT (etc) board.

See the documentation folder and examples for usage.

2 LICENSE

Copyright (c) 2015, Majenko Technologies All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Majenko Technologies nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3 Class Index

3.1 Class List

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

[A500KB](#)

[2](#)

4 Class Documentation

4.1 A500KB Class Reference

Public Member Functions

- [A500KB](#) (uint8_t clock, uint8_t data, uint8_t reset, uint8_t [drive](#)=255, uint8_t [power](#)=255)
- void [begin](#) ()
- int [scan](#) ()
- uint8_t [translate](#) (int scancode)
- void [power](#) (uint8_t state)
- void [drive](#) (uint8_t state)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 `A500KB::A500KB (uint8_t clock, uint8_t data, uint8_t reset, uint8_t drive = 255, uint8_t power = 255)`
`[inline]`

Constructor to make a new [A500KB](#) object. It takes (and stores for later use) the clock, data and reset pins the A500 keyboard is connected to. Can also take two additional pins to control the two LEDs on the keyboard.

Note: controlling the LEDs requires a 5V logic level output, so will only work with a 5V Arduino or with logic level shifting added to your system.

4.1.2 Member Function Documentation

4.1.2.1 `void A500KB::begin ()`

Initialize the keyboard system. Basically this sets the IO pins to the right direction.

4.1.2.2 `void A500KB::drive (uint8_t state)`

Control the DRIVE led. Takes a HIGH or LOW for ON or OFF.

4.1.2.3 `void A500KB::power (uint8_t state)`

Control the POWER led. Takes a HIGH or LOW for ON or OFF.

4.1.2.4 `int A500KB::scan ()`

Scan for an incoming keypress. Must be called as regularly as possible. It's (mostly) non-blocking and returns -1 if there is no keypress available. A keypress is an 8-bit value with the most significant bit set for a press and cleared for a release.

4.1.2.5 `uint8_t A500KB::translate (int scancode)`

Translate a keypress code into a character or keycode suitable for Keyboard.press() or Keyboard.release().

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

- A500KB.h

- A500KB.cpp

Index

A500KB, [2](#)
 A500KB, [2](#)
 begin, [2](#)
 drive, [2](#)
 power, [2](#)
 scan, [2](#)
 translate, [2](#)

begin
 A500KB, [2](#)

drive
 A500KB, [2](#)

power
 A500KB, [2](#)

scan
 A500KB, [2](#)

translate
 A500KB, [2](#)