```
/*
  PS2Keyboard.h - PS2Keyboard library
  Copyright (c) 2007 Free Software Foundation.  All right reserved.
  Written by Christian Weichel <info@32leaves.net>

  Modified to add F1-F12 keys and other minor corrections by Glen Popiel
- KW5GP

  ** Mostly rewritten Paul Stoffregen <paul@pjrc.com>, June 2010
  ** Modified for use with Arduino 13 by L. Abraham Smith,
<n3bah@microcompdesign.com> *
  ** Modified for easy interrup pin assignement on method
begin(datapin,irq_pin). Cuningan <cuninganreset@gmail.com> **

  This library is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.

  This library is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
  Lesser General Public License for more details.

  You should have received a copy of the GNU Lesser General Public
  License along with this library; if not, write to the Free Software
  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301
USA
*/


#ifndef PS2Keyboard_h
#define PS2Keyboard_h

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h" // for attachInterrupt, FALLING
#else
#include "WProgram.h"
#endif

// Every call to read() returns a single byte for each
// keystroke.  These configure what byte will be returned
// for each "special" key.  To ignore a key, use zero.
#define PS2_TAB                   9
#define PS2_ENTER           13
#define PS2_BACKSPACE             127
#define PS2_ESC                   27
#define PS2_INSERT                17
#define PS2_DELETE                127
#define PS2_HOME            0
#define PS2_END                   18
```

```
#define PS2_PAGEUP                25
#define PS2_PAGEDOWN              26
#define PS2_UPARROW               11
#define PS2_LEFTARROW             8
#define PS2_DOWNARROW             10
#define PS2_RIGHTARROW            21
#define PS2_F1                    5
#define PS2_F2                    6
#define PS2_F3                    4
#define PS2_F4                    12
#define PS2_F5                    3
#define PS2_F6                    2
#define PS2_F7                    14
#define PS2_F8                    15
#define PS2_F9                    1
#define PS2_F10                   16
#define PS2_F11                   19
#define PS2_F12                   7
#define PS2_SCROLL                0

#define PS2_INVERTED_EXCLAMATION  161 // ¡
#define PS2_CENT_SIGN             162 // ¢
#define PS2_POUND_SIGN            163 // £
#define PS2_CURRENCY_SIGN         164 // ¤
#define PS2_YEN_SIGN              165 // ¥
#define PS2_BROKEN_BAR                  166 // ¦
#define PS2_SECTION_SIGN          167 // §
#define PS2_DIAERESIS             168 // ¨
#define PS2_COPYRIGHT_SIGN        169 // ©
#define PS2_FEMININE_ORDINAL      170 // ª
#define PS2_LEFT_DOUBLE_ANGLE_QUOTE     171 // «
#define PS2_NOT_SIGN              172 // ¬
#define PS2_HYPHEN                173
#define PS2_REGISTERED_SIGN       174 // ®
#define PS2_MACRON                175 // ¯
#define PS2_DEGREE_SIGN                 176 // °
#define PS2_PLUS_MINUS_SIGN       177 // ±
#define PS2_SUPERSCRIPT_TWO       178 // ²
#define PS2_SUPERSCRIPT_THREE           179 // ³
#define PS2_ACUTE_ACCENT          180 // ´
#define PS2_MICRO_SIGN            181 // µ
#define PS2_PILCROW_SIGN          182 // ¶
#define PS2_MIDDLE_DOT            183 // ·
#define PS2_CEDILLA               184 // ¸
#define PS2_SUPERSCRIPT_ONE       185 // ¹
#define PS2_MASCULINE_ORDINAL           186 // º
#define PS2_RIGHT_DOUBLE_ANGLE_QUOTE    187 // »
#define PS2_FRACTION_ONE_QUARTER  188 // ¼
#define PS2_FRACTION_ONE_HALF           189 // ½
#define PS2_FRACTION_THREE_QUARTERS     190 // ¾
#define PS2_INVERTED_QUESTION MARK191 // ¿
#define PS2_A_GRAVE               192 // À
#define PS2_A_ACUTE               193 // Á
#define PS2_A_CIRCUMFLEX          194 // Â
```

```
#define PS2_A_TILDE               195 // Ã
#define PS2_A_DIAERESIS                 196 // Ä
#define PS2_A_RING_ABOVE          197 // Å
#define PS2_AE                    198 // Æ
#define PS2_C_CEDILLA             199 // Ç
#define PS2_E_GRAVE               200 // È
#define PS2_E_ACUTE               201 // É
#define PS2_E_CIRCUMFLEX          202 // Ê
#define PS2_E_DIAERESIS                 203 // Ë
#define PS2_I_GRAVE               204 // Ì
#define PS2_I_ACUTE               205 // Í
#define PS2_I_CIRCUMFLEX          206 // Î
#define PS2_I_DIAERESIS                 207 // Ï
#define PS2_ETH                   208 // Ð
#define PS2_N_TILDE               209 // Ñ
#define PS2_O_GRAVE               210 // Ò
#define PS2_O_ACUTE               211 // Ó
#define PS2_O_CIRCUMFLEX          212 // Ô
#define PS2_O_TILDE               213 // Õ
#define PS2_O_DIAERESIS                 214 // Ö
#define PS2_MULTIPLICATION        215 // ×
#define PS2_O_STROKE              216 // Ø
#define PS2_U_GRAVE               217 // Ù
#define PS2_U_ACUTE               218 // Ú
#define PS2_U_CIRCUMFLEX          219 // Û
#define PS2_U_DIAERESIS                 220 // Ü
#define PS2_Y_ACUTE               221 // Ý
#define PS2_THORN           222 // Þ
#define PS2_SHARP_S               223 // ß
#define PS2_a_GRAVE               224 // à
#define PS2_a_ACUTE               225 // á
#define PS2_a_CIRCUMFLEX          226 // â
#define PS2_a_TILDE               227 // ã
#define PS2_a_DIAERESIS                 228 // ä
#define PS2_a_RING_ABOVE          229 // å
#define PS2_ae                    230 // æ
#define PS2_c_CEDILLA             231 // ç
#define PS2_e_GRAVE               232 // è
#define PS2_e_ACUTE               233 // é
#define PS2_e_CIRCUMFLEX          234 // ê
#define PS2_e_DIAERESIS                 235 // ë
#define PS2_i_GRAVE               236 // ì
#define PS2_i_ACUTE               237 // í
#define PS2_i_CIRCUMFLEX          238 // î
#define PS2_i_DIAERESIS                 239 // ï
#define PS2_eth                   240 // ð
#define PS2_n_TILDE               241 // ñ
#define PS2_o_GRAVE               242 // ò
#define PS2_o_ACUTE               243 // ó
#define PS2_o_CIRCUMFLEX          244 // ô
#define PS2_o_TILDE               245 // õ
#define PS2_o_DIAERESIS                 246 // ö
#define PS2_DIVISION              247 // ÷
#define PS2_o_STROKE              248 // ø
```

```
#define PS2_u_GRAVE               249 // ù
#define PS2_u_ACUTE               250 // ú
#define PS2_u_CIRCUMFLEX          251 // û
#define PS2_u_DIAERESIS                252 // ü
#define PS2_y_ACUTE               253 // ý
#define PS2_thorn           254 // þ
#define PS2_y_DIAERESIS                255 // ÿ


#define PS2_KEYMAP_SIZE 136

typedef struct {
      uint8_t noshift[PS2_KEYMAP_SIZE];
      uint8_t shift[PS2_KEYMAP_SIZE];
      uint8_t uses_altgr;
      uint8_t altgr[PS2_KEYMAP_SIZE];
} PS2Keymap_t;


extern const PROGMEM PS2Keymap_t PS2Keymap_US;
extern const PROGMEM PS2Keymap_t PS2Keymap_German;


/**
 * Purpose: Provides an easy access to PS2 keyboards
 * Author:  Christian Weichel
 */
class PS2Keyboard {
  public:
      /**
       * This constructor does basically nothing. Please call the
begin(int,int)
       * method before using any other method of this class.
       */
    PS2Keyboard();

    /**
     * Starts the keyboard "service" by registering the external
interrupt.
     * setting the pin modes correctly and driving those needed to high.
     * The propably best place to call this method is in the setup
routine.
     */
    static void begin(uint8_t dataPin, uint8_t irq_pin, const PS2Keymap_t
&map = PS2Keymap_US);

    /**
     * Returns true if there is a char to be read, false if not.
     */
    static bool available();

    /**
     * Returns the char last read from the keyboard.
     * If there is no char availble, -1 is returned.
     */
    static int read();
};

// interrupt pins for known boards
#if !defined(CORE_INT0_PIN)
#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) // Arduino
Mega
#define CORE_INT0_PIN  2
#define CORE_INT1_PIN  3
#define CORE_INT2_PIN  21
#define CORE_INT3_PIN  20
#define CORE_INT4_PIN  19
#define CORE_INT5_PIN  18
#elif defined(__AVR_ATmega644P__) || defined(__AVR_ATmega644__) //
Sanguino
#define CORE_INT0_PIN  10
#define CORE_INT1_PIN  11
#define CORE_INT2_PIN  2
#else  // Arduino Duemilanove, Diecimila, LilyPad, Mini, Fio, etc...
#define CORE_INT0_PIN  2
#define CORE_INT1_PIN  3
#endif
#endif

#endif
```