

```
/******
```

```
*      comaidsystem.h
```

```
*
```

```
*      Communication Aid System:  Designed to assist on-road communication with deaf driver
```

```
*              Hardware specs: Atmega168p microcontroller
```

```
*
```

```
*      Authors: Timmy Mbaya, Brendan Davis, Joseph Cohen
```

```
*
```

```
*      Under supervision from Betty O'Neil
```

```
*
```

```
*      Spring 2010 Real-Time Systems Independent Study, UMass Boston
```

```
*
```

```
*****/
```

```
/* Copyright (c) 2010 Timmy Mbaya, Brendan Davis, Joseph Cohen
```

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 the copyright holders nor the names of 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 OWNER 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. \*/

/\* \$Id: comaidssystem.h, version 1.0 2010/31/04 09:26:08 \*/

//

```
//
```

```
//
```

```
#define KB_PCINT 11    //Pin change interrupt 11
```

```
#define SCODE_SIZE 0xF1 // Array size
```

```
#define NUM_BITS 11    // number of bits to receive (1 start, 1 parity, 1 stop)
```

```
#define BUFF_SIZE 17
```

```
#define NO_USE_CHAR ' ' //passed to functions as non-used argument
```

```
void process_scancode(unsigned char scancode);    //Function to process received scancode within  
PCINT ISR
```

```
void buffer_char(unsigned char);                //function to buffer decoded chars, but actually just sends  
them to LCD
```

```
void enable_pcint(int pcintnum);                //enables pinchange interrupts
```

```
void initTables(void);                          //initializes the scancode struct array
```

```
void init_sysvarStates(void);                   //initializes states of system global variables
```

```
void keyboard_setup(void);                      //sets up the keyboard: clears buffer
```

```
void programfn(unsigned char);                  //in program mode, buffer and programs received chars
```

```
void program(unsigned char *, unsigned char*); //in program mode, programs F-key display texts  
when called from FxFunction()
```

```
void read_eeprom_string(const uint8_t*);        //read a string from an eeprom address
```

```
//special key functions
```

```
void Fkeys_Function(unsigned char Fx_char);     //Function called by all Fx keys
```

void FxFunction(uint8\_t\* Fxtext, unsigned char\* Default\_Fxstring); //called within Fkeys\_Function based on Fx\_char

void end\_codefn(unsigned char); //EN\_CODE function

void E0fn(unsigned char); //E0 scancode function

void E1fn(unsigned char); //E1 scancode function

void bkspfn(unsigned char); //BKSP scancode function

void deletfn(unsigned char); //DELETE scancode function

void homefn(unsigned char); //HOME scancode function

void enterfn(unsigned char); //ENTER scancode function

void escapefn(unsigned char); //ESCAPE scancode function

void l\_ctrlfn(unsigned char); //LCTRL and R\_CTRL scancode function

void caplockfn(unsigned char); //CAPSLOCK scancode function

void defaultfn(unsigned char); //default scancode function: for unimplemented keys

void shiftfn(unsigned char); //SHIFT scancode function