```
#ifndef __USBAPI__
#define __USBAPI__

#if defined(USBCON)

//================================================================================
//================================================================================
//  USB

class USBDevice_
{
public:
    USBDevice_();
    bool configured();

    void attach();
    void detach();  // Serial port goes down too...
    void poll();
};
extern USBDevice_ USBDevice;

//================================================================================
//================================================================================
//  Serial over CDC (Serial1 is the physical port)

class Serial_ : public Stream
{
private:
    ring_buffer *_cdc_rx_buffer;
public:
    void begin(uint16_t baud_count);
    void end(void);

    virtual int available(void);
    virtual void accept(void);
    virtual int peek(void);
    virtual int read(void);
    virtual void flush(void);
    virtual size_t write(uint8_t);
    using Print::write; // pull in write(str) and write(buf, size) from Print
    operator bool();
};
extern Serial_ Serial;
```

```
//================================================================================
//================================================================================
//  Mouse

#define MOUSE_LEFT 1
#define MOUSE_RIGHT 2
#define MOUSE_MIDDLE 4
#define MOUSE_ALL (MOUSE_LEFT | MOUSE_RIGHT | MOUSE_MIDDLE)

class Mouse_
{
private:
    uint8_t _buttons;
    void buttons(uint8_t b);
public:
    Mouse_(void);
    void begin(void);
    void end(void);
    void click(uint8_t b = MOUSE_LEFT);
    void move(signed char x, signed char y, signed char wheel = 0);
    void press(uint8_t b = MOUSE_LEFT);     // press LEFT by default
    void release(uint8_t b = MOUSE_LEFT);   // release LEFT by default
    bool isPressed(uint8_t b = MOUSE_LEFT); // check LEFT by default
};
extern Mouse_ Mouse;

//================================================================================
// Joystick
// Implemented in HID.cpp

class Joystick_
{
public:
   Joystick_();
   void move(uint8_t x, uint8_t y, uint8_t buttons);
};
extern Joystick_ Joystick;


//================================================================================
//================================================================================
//  Keyboard

#define KEY_LEFT_CTRL       0x80
```

```
#define KEY_LEFT_SHIFT      0x81
#define KEY_LEFT_ALT        0x82
#define KEY_LEFT_GUI        0x83
#define KEY_RIGHT_CTRL      0x84
#define KEY_RIGHT_SHIFT     0x85
#define KEY_RIGHT_ALT       0x86
#define KEY_RIGHT_GUI       0x87

#define KEY_UP_ARROW        0xDA
#define KEY_DOWN_ARROW      0xD9
#define KEY_LEFT_ARROW      0xD8
#define KEY_RIGHT_ARROW     0xD7
#define KEY_BACKSPACE       0xB2
#define KEY_TAB             0xB3
#define KEY_RETURN          0xB0
#define KEY_ESC             0xB1
#define KEY_INSERT          0xD1
#define KEY_DELETE          0xD4
#define KEY_PAGE_UP         0xD3
#define KEY_PAGE_DOWN       0xD6
#define KEY_HOME            0xD2
#define KEY_END             0xD5
#define KEY_CAPS_LOCK       0xC1
#define KEY_F1              0xC2
#define KEY_F2              0xC3
#define KEY_F3              0xC4
#define KEY_F4              0xC5
#define KEY_F5              0xC6
#define KEY_F6              0xC7
#define KEY_F7              0xC8
#define KEY_F8              0xC9
#define KEY_F9              0xCA
#define KEY_F10             0xCB
#define KEY_F11             0xCC
#define KEY_F12             0xCD

//  Low level key report: up to 6 keys and shift, ctrl etc at once
typedef struct
{
    uint8_t modifiers;
    uint8_t reserved;
    uint8_t keys[6];
} KeyReport;

class Keyboard_ : public Print
```

```
{
private:
    KeyReport _keyReport;
    void sendReport(KeyReport* keys);
public:
    Keyboard_(void);
    void begin(void);
    void end(void);
    virtual size_t write(uint8_t k);
    virtual size_t press(uint8_t k);
    virtual size_t release(uint8_t k);
    virtual void releaseAll(void);
};
extern Keyboard_ Keyboard;


//================================================================================
//================================================================================
//  Low level API

typedef struct
{
    uint8_t bmRequestType;
    uint8_t bRequest;
    uint8_t wValueL;
    uint8_t wValueH;
    uint16_t wIndex;
    uint16_t wLength;
} Setup;

//================================================================================
//================================================================================
//  HID 'Driver'

int     HID_GetInterface(uint8_t* interfaceNum);
int     HID_GetDescriptor(int i);
bool    HID_Setup(Setup& setup);
void    HID_SendReport(uint8_t id, const void* data, int len);

//================================================================================
//================================================================================
//  MSC 'Driver'

int     MSC_GetInterface(uint8_t* interfaceNum);
int     MSC_GetDescriptor(int i);
```

```
bool    MSC_Setup(Setup& setup);
bool    MSC_Data(uint8_t rx,uint8_t tx);

//================================================================================
//================================================================================
//  CSC 'Driver'

int     CDC_GetInterface(uint8_t* interfaceNum);
int     CDC_GetDescriptor(int i);
bool    CDC_Setup(Setup& setup);

//================================================================================
//================================================================================

#define TRANSFER_PGM        0x80
#define TRANSFER_RELEASE    0x40
#define TRANSFER_ZERO       0x20

int USB_SendControl(uint8_t flags, const void* d, int len);
int USB_RecvControl(void* d, int len);

uint8_t USB_Available(uint8_t ep);
int USB_Send(uint8_t ep, const void* data, int len);    // blocking
int USB_Recv(uint8_t ep, void* data, int len);      // non-blocking
int USB_Recv(uint8_t ep);                           // non-blocking
void USB_Flush(uint8_t ep);


//================================================================================


#endif

#endif /* if defined(USBCON) */
```