Debug with aid of SPI interface

Usage

- create a directory 'debugger' with 'debugger.h' in it
- zip the directory to debugger.zip
- //include library/add ZIP library within Arduino IDE
- //include library/debugger.h
- use the example to test
- documentation of SPI slave with LEONARDOs see kner.at

Example

```
/* example for debug.h
*  caveat: include SPI.h before include debugger.h
*
*/

#define DEBUG
#include <SPI.h>
#include <debugger.h>
void setup() {
 beginDebug ();
}

void loop() {
 TraceFunc (); // where am i?
 Trace(F("here i am "));
 Traceln2(13,HEX);
 delay(1000);
}
```

debugger.h

```
// Written by Nick Gammon
// September 2011

#ifndef Debugger_h
#define Debugger_h
#define SSS 13   //this pin number must be adapted to the external line
connected to the spi master

#ifdef DEBUG
 #define beginDebug()  do { SPI.begin (); SPI.setClockDivider(SPI_CLOCK_DIV8); }
while (0)
 #define Trace(x)      SPIdebug.print   (x)
 #define Trace2(x,y)   SPIdebug.print   (x,y)
 #define Traceln(x)    SPIdebug.println (x)
 #define Traceln2(x,y) SPIdebug.println (x,y)
 #define TraceFunc()   do { SPIdebug.print (F("In function: "));
SPIdebug.println (__PRETTY_FUNCTION__); } while (0)

 class tSPIdebug : public Print
```

```
 {
 public:
   virtual size_t write (const byte c)
     {
     digitalWrite(SSS, LOW);
     SPI.transfer (c);
     digitalWrite(SSS, HIGH);
     return 1;
     }  // end of tSPIdebug::write
 }; // end of tSPIdebug

 // an instance of the SPIdebug object
 tSPIdebug SPIdebug;

#else
 #define beginDebug()  ((void) 0)
 #define Trace(x)      ((void) 0)
 #define Trace2(x,y)   ((void) 0)
 #define Traceln(x)    ((void) 0)
 #define Traceln2(x,y) ((void) 0)
 #define TraceFunc()   ((void) 0)
#endif

#endif
```