SPI protocol for ATmega328pb/p

Created primarly for SD card module and to make it maximum lightweight.

main features

- included `library.json` so can be easily attached to other project with platformio library manager. Just add to `platformio.ini` the line:

```
lib_deps = https://github.com/328pb/spi
```

- minimum implementation to make it lightweight

```
AVR Memory Usage
----------------
Device: Unknown

Program:     518 bytes
(.text + .data + .bootloader)

Data:          0 bytes
(.data + .bss + .noinit)
```

- no Arduino libs dependency
- only one interface SPI0 is used
- library do not use interrupts, just loop until response available

usage

Library implements a class SPI::SPI() and provides following methods:

- SPI::init() - initialize with default speed and address (see i2c.h)
- SPI::init(uint8_t clock_div);
- SPI::init(uint8_t clock_div, uint8_t cs, uint8_t auto_cs) - initialize with given speed, chip selection pin and CS pin logic (see below). **SPEED IS DEFINED AS CPU FREQ DIVIDER**
- SPI::send_ln(uint8_t *data, uint8_t len) - sends data in 8bit chunks `len` times.
- SPI::send(uint8_t data) - send single byte
- SPI::cs_on() - chip select (pin pull low)
- SPI::cs_off() - chip unselect (pin pull up)
- SPI::off() - turn off SPI and restore back SPI interrupts setting

CS selection pin logic is defined by auto_cs argument in init() method. When set to `true` CS pin will be togled at each data transmission (byte if send() or whole data if send_ln())

example

Script sends sample text, no need for SPI device, just network analyzer. Compile examples/main.cpp ([env:demo] in platformio.ini).