I2C protocol for ATmega328pb/p

Created primarly for OLED/LCD display 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/i2c
```

- minimum implementation to make it lightweight

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

Program:     754 bytes
(.text + .data + .bootloader)

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

- no Arduino libs dependency
- only Master Transmition
- on ATmega328pb uses only TWI0
- library do not use interrupts, just loop until response available

usage

Library implements a class I2C::I2C() and provides following methods:

- I2C::init() - initialize with default speed and address (see i2c.h)
- I2C::init(uint8_t address, uint16_t speed) - initialize with given address and speed. **SPEED IS IN kHz**
- I2C::send_ln(uint8_t *data, uint8_t len) - sends data in 8bit chunks `len` times.
- I2C::send(uint8_t data) - send single byte
- I2C::off() - turn off TWI and restore back TWI interrupts setting
- I2C::test- when set `true` will not wait for ACK from slave. By default `false`

Class will send stop signal after each transmission unless test=true (see below). On DEV_ERR error, the TWI module will be turned off and interrupts restored (needs init() to reinitialize).

Class also exposes I2C.err variable of type error:

```
typedef enum : uint8_t {
  NO_ERR = 0,
  DEV_ERR = 1,
  COM_ERR = 2,
} error;
```

When library compiled with ENABLE_I2C_SCAN directive, aditional method uint8_t I2C::scan() will be available. The method will scan through all addresses and set one when recive ACK. Return address on success or 0. Do not forget to init() after setting new address. In most cases not needed and waste memory (aditional 114 bytes), but can be usefull in rare cases when device address is not known.

example

Script sends sample text with `test=true`, so no need for I2C device, just network analyzer. Compile examples/main.cpp ([env:demo] in platformio.ini).