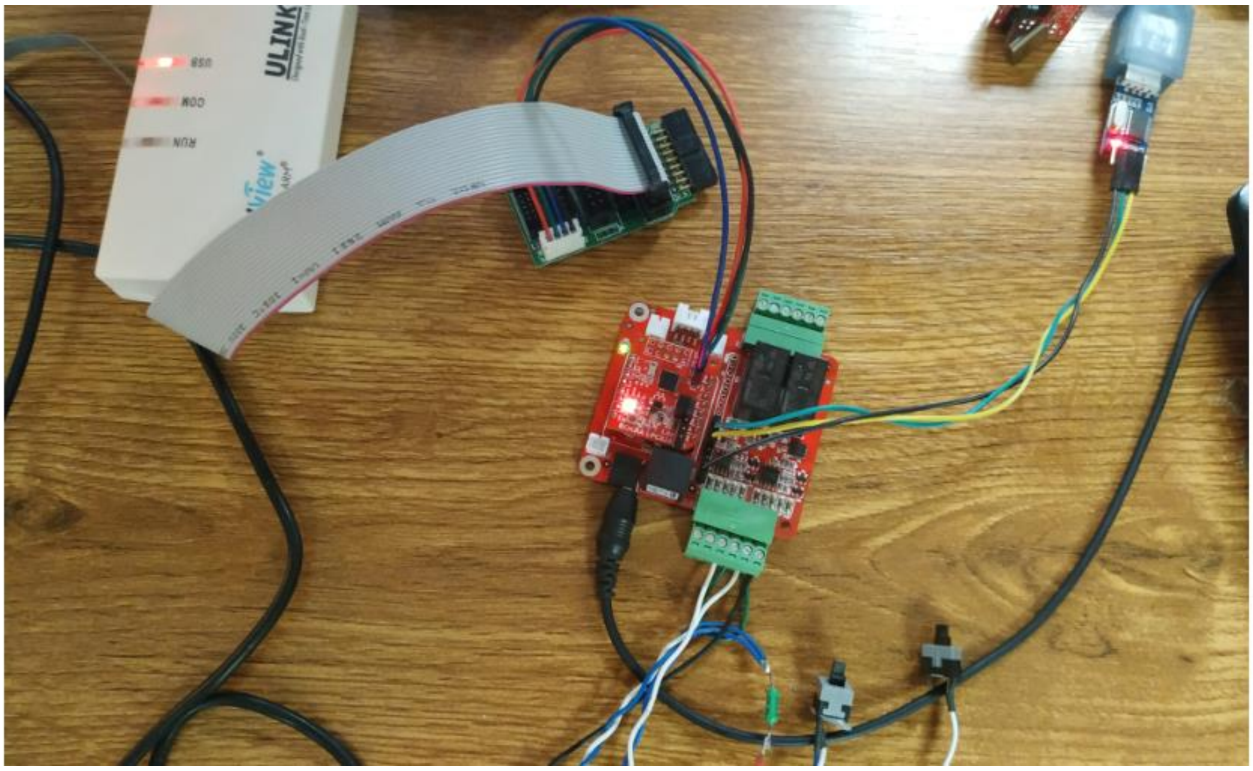


BOKRA485

Firmware manual

V1.10

2020.01.25

Power Up Sequence

The power up sequence of the MCU is same as the reset MCU sequence.
Firstly it is readed a preddefined MCU clock registers and recalculate it system and the main clock. This needs is for the HAL library.
Next is initialized the MCU HW, board and peripherials.
MCU needs initialize the GPIO pins to default values with aspect to the connected hardware (UART, I2C, LEDs, Crystal,...) without any physical impacts (LED blinks, I2C stuck,...). This is done in Board_MCU_Pin_Init().

Next, MCU Clock subsystem is switched to use external crystal as a main clock source. Is used also internal PLL for destination clock 30MHz for the peripherial, and 60MHz for CPU.
If external crystal isn't present or is fail, program will halt here.

After these critical actions, is initialized MCU's UART and I2C block.

Global buffers and application structure are initialized to the zero or depends on application needs in App_Init()

I2C GPIO expander MCU23008 is initialized depends on this application requirements. 8-bit GPIO are divided to half and GPIO 0-3 are used as inputs, GPIO 4-7 are used as outputs.
Note: IOPOL register for inputs 0-3 is preset for inverse polarity, because hardware is designed as is (this is not failure).
At the last are enabled interrupts and program go to main loop.
Interrupt is used for UART communication, TickTimer (1/1000 of second) and I2C communication.

Boot sequence

After power-up board (or after reset), firmware flow is this:
1. Initializing the system clock and board [in this routine, program can stuck only if microprocessor is bad/wrong, or external crystal don't oscillate]
2. Initializing application and stopwatch [0% chance to stuck program]
3. Switching ON the Green LED [start of first 0.5sec blink]
4. Initializing of the MCP23008 and I2C [program can be stuck, if wrong I2C address are selected by HW (A0, A1)]
5. Waiting for 0.5sec
6. Switching OFF the Green LED [end of first 0.5sec blink]
7. Waiting for 0.5sec
8. Enabling UART interrupt [program can hardly slower if UART communication is active with high data payloads - Master error]
9. Switching ON the Green LED [start of second 0.5sec blink]
10. Waiting for 0.5sec
11. Switching OFF the Green LED [end of second 0.5sec blink]
12. Run to main loop

Main loop

Program in the main loop periodically check presence of UART received character, and ScanLogic_Flag (this is set each 1/100 of second), Second_flag (communication lost count down timer) and request for I2C write to MCP23008 expander.

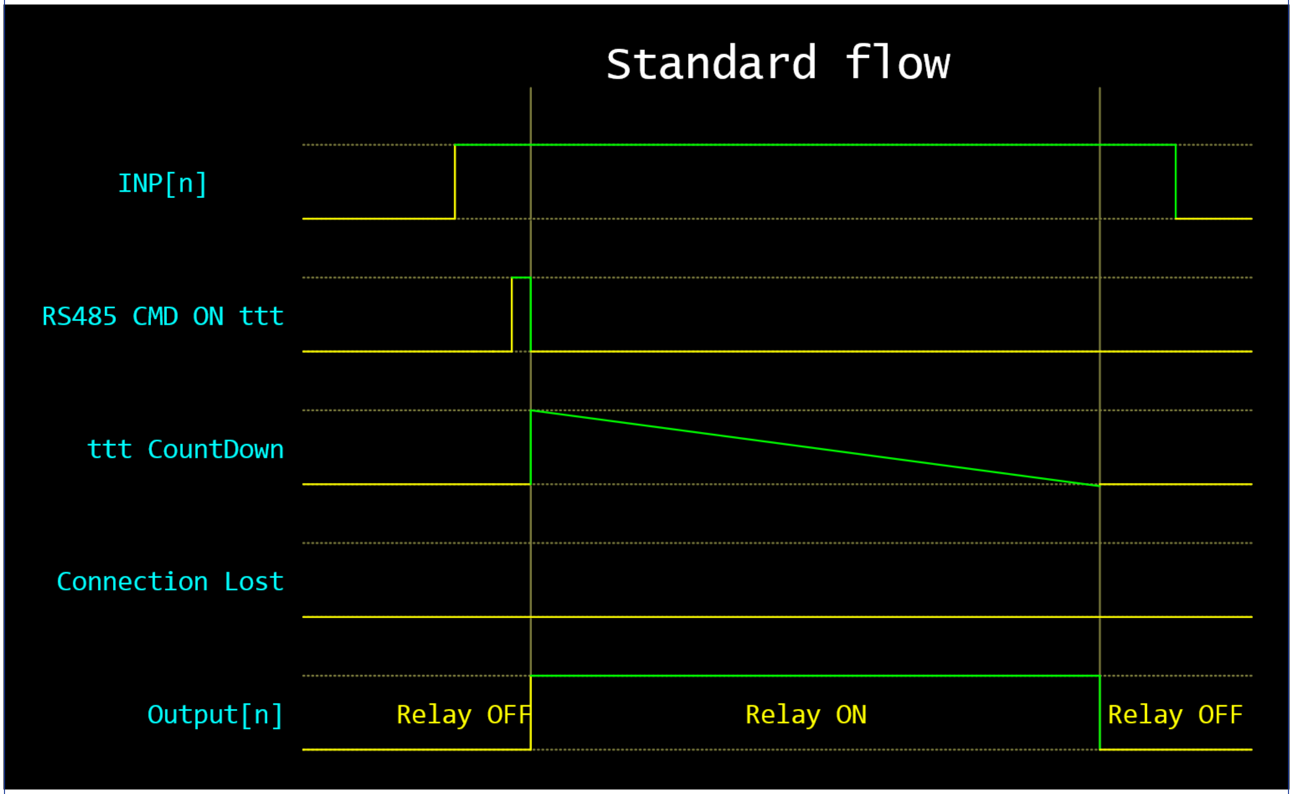
If ScanLogic_Flag is set, application flow is checked and required action are run.

In general, application is waiting for any inputs. In depends on their, is action called.

Inputs are:
- GPIO pins on BOKRA I2C 2DI+2DO board (IN1 and IN2)
- UART commands
- timeouts if set

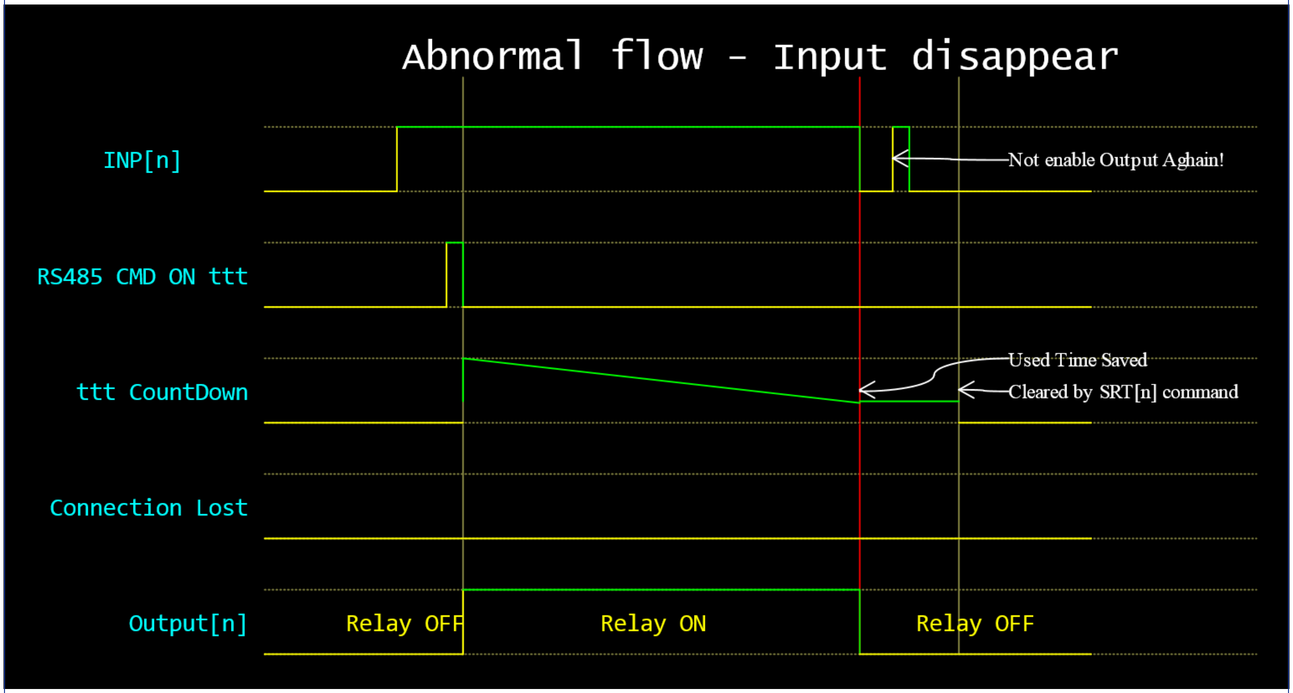
Outputs are:
- GPIO pins on BOKRA I2C 2DI+2DO board (OUT1 and OUT2)
- reload values for timeouts
- UART response codes

Standard application flow



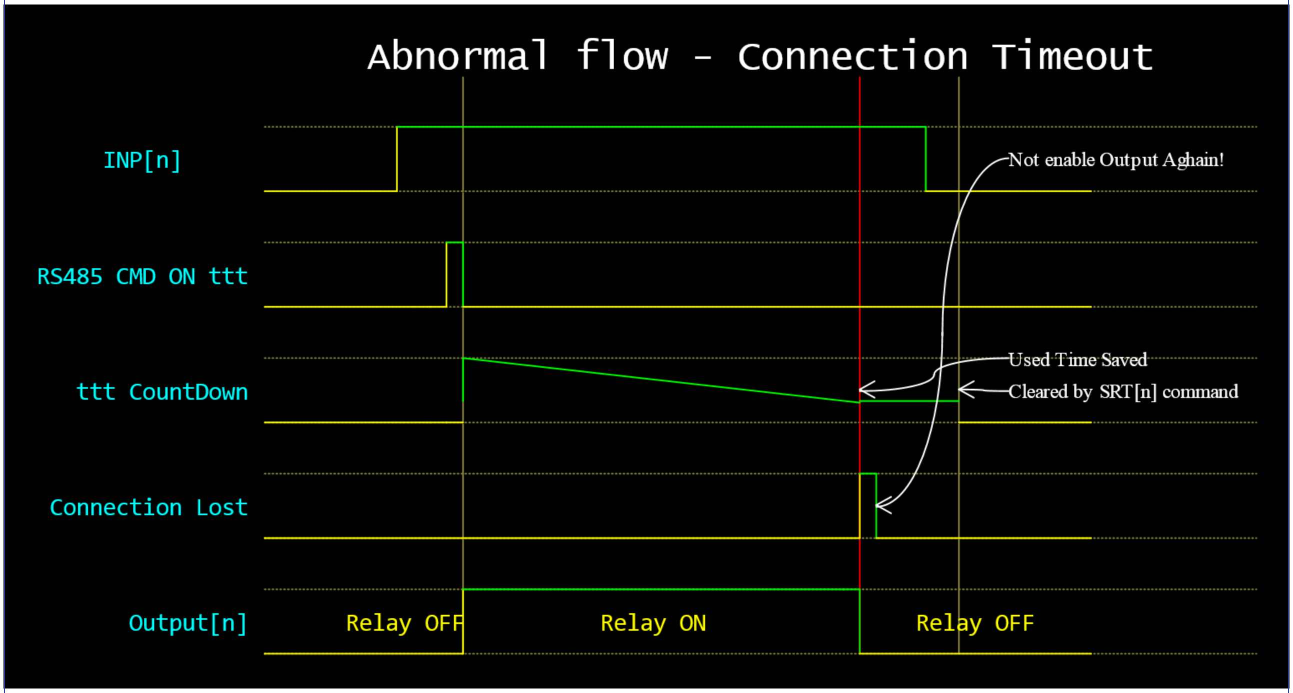
In this case, application after receiving ON command wait for INP physical signal (IN1 or IN2 on 4DI+4DO board). After presence this signal, is started countdown timer (ttt value is timeout time) and output signal is activated (activate relay).
If countdown timer is reached, output signal is deactivated (Relay OFF).

Abnormal situation – Input signal disappear



In this case is Output deactivated before timeout count from ttt value to zero. After output switch OFF, time value of Output active time is saved. This value is send via UART after receiving request SRT.
Value is cleared after response to SRT command, or after receiving a new ON command.

Abnormal situation – connection timeout



In this case is output deactivated after connection lost appear.
Connection lost is set if Alive_CountDown (in App structure) reach zero value. This variable is reloaded in receiving ASK command, or ADO command in output active stage or after reset.
Manipulation with active output time is same as in previous situation.

UART Command decoder

Command decoder is very simple. Because commands are text strings, each received character is stored in RxCMD array (App structure) and if received character is /0, decoder are started.
Decoder work simple on string comparate principe.
If received string is equal to constant, action will called. Otherwise is cleared.

All commands are fired by receiving last zero character!

Uart communication parameters are: 9600 baud, 8bits, 1 stop bit, without parity

I2C Communication

I2C is preconfigured for 400kHz data rate, and used fix device address of MCP23008. Address is stored in Config.h file and after eventually changes project have to recompiled and updated into MCU.
Of course, this address must meet the address preselected by HW (A0, A1 and A2 settings by JP25-JP27).
If address is different, Green LED L1 on MCU module staying lit ON and program are stopped.

Indicators

Two assembled LED on LPC824 Lite board are used as application status

LED L1 – Green
In the boot process:
Blink twice (0.5sec) after initialize all required blocks.

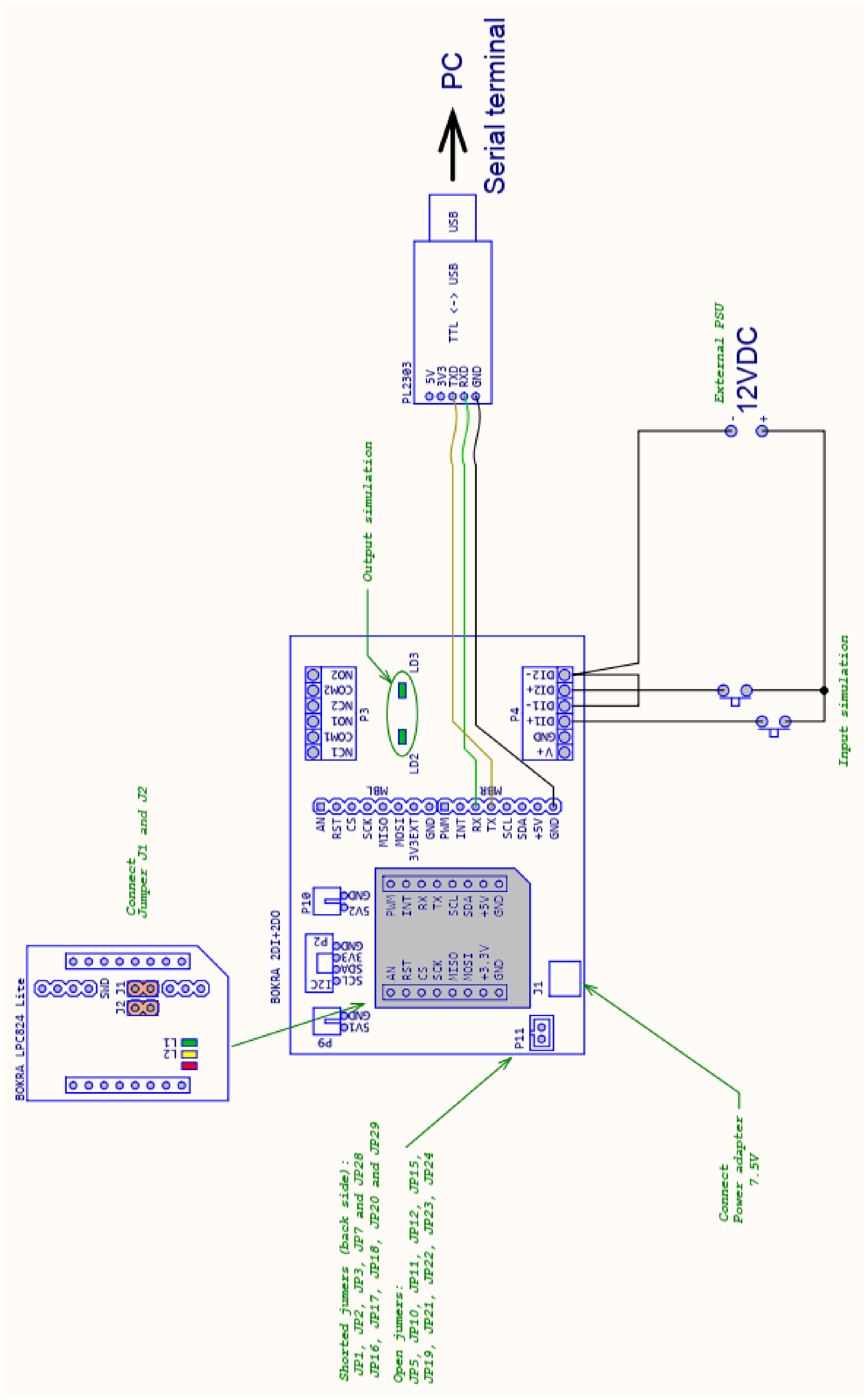
Based on boot flow, if I2C communication freeze (bad I2C address, missing jumper J1 or J2 on LPC824 MCU module, ...) first blink don't finish, and Green LED still lit On and program is stopped.

If first blink is gone, second blink can be unfinished only when UART communication is active in high bandwidth (master continuosly send commands into module and don't wait to response).

If application is running:
Blink (100ms) after receive character from UART.

LED L2 – Yellow is used as UART received commands decoder indicator. LED blink if know command was received and decoded.

Debug connection ans settings


PC
Serial terminal
USB
TTL <-> USB
PL2303
5V
3V3
TXD
RXD
GND
External PSU
12VDC
Output simulation
Input simulation
LD2
LD3
P3
P4
NC1
COM1
NO1
NC2
COM2
NO2
V+
GND
DI1+
DI1-
DI2+
DI2-
MBL
MBR
AN
RST
CS
SCK
MISO
MOSI
3V3EXT
GND
PWM
INT
RX
TX
SCL
SDA
+5V
GND
+3.3V
P10
P2
P9
P11
J1
I2C
Connect Jumper J1 and J2
BOKRA LPC824 Lite
SWD
J1
J2
L1
L2
BOKRA 2DI+2DO
Shorted jumers (back side):
JP1, JP2, JP3, JP7 and JP28
JP16, JP17, JP18, JP20 and JP29
Open jumers:
JP5, JP10, JP11, JP12, JP15,
JP19, JP21, JP22, JP23, JP24
Connect Power adapter 7.5V