commentrgb0.5,0.0,0.0 keywordrgb0.0,0.5,0.0 keywordtypergb0.38,0.25,0.125 keywordflowrgb0.88,0.5,0.0 preprocessorrgb0.5,0.38,0.125 stringliteralrgb0.0,0.125,0.25 charliteralrgb0.0,0.5,0.5 vhdldigitrgb1.0,0.0,1.0 vhdlkeywordrgb0.43,0.0,0.43 vhdl-logicrgb1.0,0.0,0.0 vhdlcharrgb0.0,0.0,0.0

darkgray

analizer

0.1

Generated by Doxygen 1.8.7

Sat Mar 11 2017 12:18:15

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Module Documentation

4.1 Main menu

Functions

- void loop ()

4.1.1 Detailed Description

4.1.2 Function Documentation

4.1.2.1 void loop ( )

Infinite cycle Will perform after startup device Available in main menu commands:

- **q**
  Go to factory calibration function.
- **e**
  Express calibration method. Not implemented yet
- **s**
  One mode of measurements with option of coeffs calibration
  Manually choosing preamps parameters
  Sample&Holde mode
- **m**
  One mode of measurements with option of coeffs calibration
  Manually choosing preamps parameters
  Inpulse measurements mode
- **a**
  One mode of measurements with option of coeffs calibration
  Series of measurements with predefined preamp parameters
  Sample&Holde mode

Definition at line 81 of file analizer.cpp.

4.2 Calibration mode

Functions

- void factoryCalibr ()

4.2.1 Detailed Description

4.2.2 Function Documentation

4.2.2.1 void factoryCalibr ( )

Main calibration function Available in calibration menu commands:

- **w**
  Set width of impulse in microseconds
  step is 1us, max 200us
- **t**
  Save to eeprom value of input resistanse of second OpAmp
- **a**
  Set currents for LED in channel **1** Max value is 1000mA, with step 1mA
- **b**
  Set currents for LED in channel **2** Max value is 1000mA, with step 1mA
- **g**
  reset LEDs if something wrong
- **f**
  Light-up first led
- **n**
  Next led choose
- **o**
  Start infinite series of pulse to LED
- **p**
  Make one pulse to LED
- **i**
  Disable LED
- **z**
  Return current led number
- **c**
  Set coeffs of OpAmp
  Control digital potentiometers ad5141
  Max value is 100kOhm, with step 0.39kOhm

- **r**
  Go to preAmp calibration

- **s**
  Save parameters for all LED to EEPROM

- **e**
  Exit from calibration mode

Definition at line 149 of file analizer.cpp.

4.3 Etalon's data write

Functions

- void preAmpCalibr ()

4.3.1 Detailed Description

4.3.2 Function Documentation

4.3.2.1 void preAmpCalibr ( )

Amplifier calibration function Available in PreAmp calibration menu commands:

- **c**
  Set coeffs of OpAmp
  Control digital potentiometers ad5141
  Max value is 100kOhm, with step 0.39kOhm

- **x**
  Choose etalon cell in memory

- **e**
  Exit from calibration mode

- **s**
  Save parameters to EEPROM

- **z**
  Return current etalon cell number

- **k**
  Save k for etalon

- **m**
  set min and max value of g2

- **n**
  Compute Ai

- **a**
  Compute Ai with doMeasurementsSH_Avg method

- **l**
  Write Ai in manual mode

Definition at line 374 of file analizer.cpp.

Chapter 5

Class Documentation

5.1 current_t Struct Reference

```
#include <analizer.h>
```

Public Attributes

- uint16_t curr1
- uint16_t curr2

5.1.1 Detailed Description

Definition at line 65 of file analizer.h.

5.1.2 Member Data Documentation

5.1.2.1 uint16_t current_t::curr1

Definition at line 67 of file analizer.h.

5.1.2.2 uint16_t current_t::curr2

Definition at line 68 of file analizer.h.

The documentation for this struct was generated from the following file:

- /home/zvebabi/Documents/workspace/LEDMicrosensor_project0/analizer/analizer.h

5.2 etalon_t Struct Reference

```
#include <analizer.h>
```

Public Attributes

- float k [NUM_OF_LED]
- float g1

- float g2mid
- float g2min
- float g2max

5.2.1 Detailed Description

Definition at line 53 of file analizer.h.

5.2.2 Member Data Documentation

5.2.2.1 float etalon_t::g1

Definition at line 56 of file analizer.h.

5.2.2.2 float etalon_t::g2max

Definition at line 59 of file analizer.h.

5.2.2.3 float etalon_t::g2mid

Definition at line 57 of file analizer.h.

5.2.2.4 float etalon_t::g2min

Definition at line 58 of file analizer.h.

5.2.2.5 float etalon_t::k[NUM_OF_LED]

Definition at line 55 of file analizer.h.

The documentation for this struct was generated from the following file:

- /home/zvebabi/Documents/workspace/LEDMicrosensor_project0/analizer/analizer.h

Chapter 6

File Documentation

6.1 /home/zvebabi/Documents/workspace/LEDMicrosensor_project0/analizer/analize File Reference

```
#include "analizer.h"
```

Functions

- ISR (TIMER1_COMPA_vect)
- int main (void)
- void setup ()
- void loop ()
- void factoryCalibr ()
- void preAmpCalibr ()
- void doMeasurements (uint8_t numOfEtalon, bool calcNorm)
- void doMeasurementsSH (uint8_t numOfEtalon, bool calcNorm)
- void doMeasurementsSH_Avg (bool calcNorm)
- void readADCOneTime (uint16_t &value)
- void readADC (float &value)
- void writeConfigToUart ()
- void setC_R (float val)
- void doOnePulse (uint16_t pulseWidth)
- void dischargeSampleHold ()
- void shiftRegisterReset ()
- void shiftRegisterNext ()
- void shiftRegisterFirst ()
- void disableLED ()
- void SerialClean ()
- void setPreAmp (float RWB1, float RWB2)
- void setCurrent (uint8_t channelN, uint16_t curValue)
- void setPulseWidth (uint16_t width)

6.1.1 Function Documentation

6.1.1.1 void disableLED ( )

Utility function for calibration mode

Definition at line 983 of file analizer.cpp.

6.1.1.2 void dischargeSampleHold ( )

Discharge capasitor, while measurements going in Sample&Hold mode

Definition at line 923 of file analizer.cpp.

6.1.1.3 void doMeasurements (uint8_t *numOfEtalon = `0`,* bool *calcNorm = `false`*)

One mode of measurements with option of coeffs calibration Manually choosing preamps parameters Inpulse measurements

Parameters

|>p0.10|>p0.15|p0.678|

`in` *numOfEtalon*	- which etalon we use for fast calibration
`in` *calcNorm*	- true - calibration, false - measurements

Definition at line 568 of file analizer.cpp.

6.1.1.4 void doMeasurementsSH (uint8_t *numOfEtalon = `0`,* bool *calcNorm = `false`*)

One mode of measurements with option of coeffs calibration Manually choosing preamps parameters Sample&Holde mode

Parameters

|>p0.15|p0.805|

numOfEtalon	- which etalon we use for fast calibration
calcNorm	- true - calibration, false - measurements

Definition at line 648 of file analizer.cpp.

6.1.1.5 void doMeasurementsSH_Avg (bool *calcNorm = `false`*)

One mode of measurements with option of coeffs calibration Series of measurements with predefined preamp parameters Sample&Holde mode

Parameters

|>p0.15|p0.805|

calcNorm	- true - calibration, false - measurements

Definition at line 728 of file analizer.cpp.

6.1.1.6 void doOnePulse (uint16_t *pulseWidth*)

Make one pulse to the LED with previously setted current and time of pulse

Definition at line 915 of file analizer.cpp.

6.1.1.7 ISR (TIMER1_COMPA_vect)

Analizer.cpp

Author

LED Microsensor

Definition at line 8 of file analizer.cpp.

6.1.1.8 int main (void)

Definition at line 15 of file analizer.cpp.

6.1.1.9 void readADC (float & *value*)

Utility function for Sample&Holde mode of measurements make 16 samples and calc average,

Parameters

|>p0.10|>p0.15|p0.678|

`out` *value* buffer for reading voltage (mV)

Definition at line 823 of file analizer.cpp.

6.1.1.10 void readADCOneTime (uint16_t & *value*) `[inline]`

Utility function for inpulse mode of measurements

Parameters

|>p0.10|>p0.15|p0.678|

`out` *value* buffer for reading adc value (w/o convert to mV)

Definition at line 814 of file analizer.cpp.

6.1.1.11 void SerialClean ( )

Utility function for cleanserial port buffer

Definition at line 991 of file analizer.cpp.

6.1.1.12 void setC_R (float *val*)

Save to eeprom value of input resistanse of second OpAmp

Parameters

|>p0.15|p0.805|

val

Definition at line 910 of file analizer.cpp.

6.1.1.13 void setCurrent (uint8_t *channelN,* uint16_t *currValue*)

This function set currents for LED in channel channelN Control DAC Max value is 1000mA, with step 1mA After current will set, it send a message with currently setted values to serial port

Parameters

|>p0.15|p0.805|

channelN	- Number of channel 1 or 2
currValue	- DAC output voltage, max 1000mV it is equal 1000mA on LED

Definition at line 1044 of file analizer.cpp.

6.1.1.14 void setPreAmp (float *RWB1,* float *RWB2*)

This function set coeffs of OpAmp Control digital potentiometers ad5141 Max value is 100kOhm, with step 0.39k↩
Ohm After resistance will set, it send a message with currently setted values to serial port

Parameters

|>p0.15|p0.805|

RWB1	1st cascade (current to voltage)
RWB2	2nd cascade (signal amplification)

Definition at line 1004 of file analizer.cpp.

6.1.1.15 void setPulseWidth (uint16_t *width*)

Set width of impulse in microseconds step is 1us, max 200us

Parameters

|>p0.15|p0.805|

width	- time of pulse, max 200us

Definition at line 1089 of file analizer.cpp.

6.1.1.16 void setup ( )

Main initialization of MCU

Definition at line 27 of file analizer.cpp.

6.1.1.17 void shiftRegisterFirst ( )

Utility function for work with shift register Select first pair of led

Definition at line 970 of file analizer.cpp.

6.1.1.18 void shiftRegisterNext ( )

Utility function for work with shift register Select next led to ON

Definition at line 959 of file analizer.cpp.

6.1.1.19 void shiftRegisterReset ( )

Utility function for work with shift register Set all outputs to High level

Definition at line 934 of file analizer.cpp.

6.1.1.20 void writeConfigToUart ( )

Print all saved to eeprom data to serial port

Definition at line 843 of file analizer.cpp.

6.2 /home/zvebabi/Documents/workspace/LEDMicrosensor_project0/analizer/analize File Reference

```
#include "Arduino.h"
#include <stdio.h>
#include <util/delay.h>
#include <avr/eeprom.h>
#include <SPI.h>
```

Classes

- struct etalon_t
- struct current_t

Macros

- #define GeneratorPin 9
- #define ShiftRegisterDelay 1
- #define DACDelay 1
- #define WBDelay 1000
- #define DISCHARGE_DELAY 50
- #define PULSE_DELAY 25
- #define REFERENCE_V 3000.0
- #define NUM_OF_LED 45
- #define NUM_OF_ETALON 3
- #define MAX_PULSE_WIDTH 150
- #define MAX_CURRENT 1000
- #define ADC_PORT PORTE
- #define DAC_PORT PORTF
- #define PREAMP_PORT PORTB
- #define SS_ADC PE6
- #define SS_PREAMP PB4
- #define SS_DAC PF1
- #define SR_ENABLE PB7
- #define SR_CLR PD4
- #define SR_DATA PD7
- #define SR_CLK PD6
- #define SH_PORT PORTD
- #define SH_SET PD0
- #define SH_RESET PD1
- #define GEN1 OCR1A
- #define GEN2 OCR1B

Functions

- void setPreAmp (float RWB1, float RWB2)
- void setCurrent (uint8_t channelN, uint16_t currValue)
- void setPulseWidth (uint16_t width)
- void setC_R (float val)
- void doOnePulse (uint16_t pulseWidth)
- void dischargeSampleHold ()
- void disableLED ()
- void factoryCalibr ()
- void preAmpCalibr ()
- void doMeasurementsSH (uint8_t numOfEtalon=0, bool calcNorm=false)
- void doMeasurementsSH_Avg (bool calcNorm=false)
- void doMeasurements (uint8_t numOfEtalon=0, bool calcNorm=false)
- void readADCOneTime (uint16_t &value)
- void readADC (float &value)
- void writeConfigToUart ()
- void shiftRegisterReset ()
- void shiftRegisterNext ()
- void shiftRegisterFirst ()
- void SerialClean ()
- void setup ()
- void loop ()
- void __cxa_pure_virtual ()

Variables

- uint8_t EEMEM _empty [20] = {0xF}
- uint16_t EEMEM _pulseWidth = 80
- float EEMEM _c_R = 3.9
- current_t EEMEM _pairsOfCurrent [NUM_OF_LED]
- etalon_t EEMEM _etalons [NUM_OF_ETALON]
- float EEMEM _coefficients [NUM_OF_LED]
- volatile current_t cur4AllLed [NUM_OF_LED]
- volatile float coeffs [NUM_OF_LED]
- volatile bool oneTimes = false
- volatile bool pulseEnd =false

6.2.1 Macro Definition Documentation

6.2.1.1 #define ADC_PORT PORTE

Definition at line 28 of file analizer.h.

6.2.1.2 #define DAC_PORT PORTF

Definition at line 29 of file analizer.h.

6.2.1.3 #define DACDelay 1

Definition at line 17 of file analizer.h.

6.2.1.4 #define DISCHARGE_DELAY 50

Definition at line 19 of file analizer.h.

6.2.1.5 #define GEN1 OCR1A

Definition at line 47 of file analizer.h.

6.2.1.6 #define GEN2 OCR1B

Definition at line 48 of file analizer.h.

6.2.1.7 #define GeneratorPin 9

Analizer.cpp

Author

LED Microsensor

Definition at line 15 of file analizer.h.

6.2.1.8 #define MAX_CURRENT 1000

Definition at line 25 of file analizer.h.

6.2.1.9 #define MAX_PULSE_WIDTH 150

Definition at line 24 of file analizer.h.

6.2.1.10 #define NUM_OF_ETALON 3

Definition at line 23 of file analizer.h.

6.2.1.11 #define NUM_OF_LED 45

Definition at line 22 of file analizer.h.

6.2.1.12 #define PREAMP_PORT PORTB

Definition at line 30 of file analizer.h.

6.2.1.13 #define PULSE_DELAY 25

Definition at line 20 of file analizer.h.

6.2.1.14 #define REFERENCE_V 3000.0

Definition at line 21 of file analizer.h.

6.2.1.15 #define SH_PORT PORTD

Definition at line 42 of file analizer.h.

6.2.1.16 #define SH_RESET PD1

Definition at line 44 of file analizer.h.

6.2.1.17 #define SH_SET PD0

Definition at line 43 of file analizer.h.

6.2.1.18 #define ShiftRegisterDelay 1

Definition at line 16 of file analizer.h.

6.2.1.19 #define SR_CLK PD6

Definition at line 39 of file analizer.h.

6.2.1.20 #define SR_CLR PD4

Definition at line 37 of file analizer.h.

6.2.1.21 #define SR_DATA PD7

Definition at line 38 of file analizer.h.

6.2.1.22 #define SR_ENABLE PB7

Definition at line 36 of file analizer.h.

6.2.1.23 #define SS_ADC PE6

Definition at line 31 of file analizer.h.

6.2.1.24 #define SS_DAC PF1

Definition at line 33 of file analizer.h.

6.2.1.25 #define SS_PREAMP PB4

Definition at line 32 of file analizer.h.

6.2.1.26 #define WBDelay 1000

Definition at line 18 of file analizer.h.

6.2.2 Function Documentation

6.2.2.1 void __cxa_pure_virtual ( )

Definition at line 224 of file analizer.h.

6.2.2.2 void disableLED ( )

Utility function for calibration mode

Definition at line 983 of file analizer.cpp.

6.2.2.3 void dischargeSampleHold ( )

Discharge capasitor, while measurements going in Sample&Hold mode

Definition at line 923 of file analizer.cpp.

6.2.2.4 void doMeasurements (uint8_t *numOfEtalon* = `0`, bool *calcNorm* = `false`)

One mode of measurements with option of coeffs calibration Manually choosing preamps parameters Inpulse measurements

Parameters

|>p0.10|>p0.15|p0.678|

`in`	*numOfEtalon*	- which etalon we use for fast calibration
`in`	*calcNorm*	- true - calibration, false - measurements

Definition at line 568 of file analizer.cpp.

6.2.2.5 void doMeasurementsSH (uint8_t *numOfEtalon* = `0`, bool *calcNorm* = `false`)

One mode of measurements with option of coeffs calibration Manually choosing preamps parameters Sample&Holde mode

Parameters

|>p0.15|p0.805|

numOfEtalon	- which etalon we use for fast calibration
calcNorm	- true - calibration, false - measurements

Definition at line 648 of file analizer.cpp.

6.2.2.6 void doMeasurementsSH_Avg (bool *calcNorm* = `false`)

One mode of measurements with option of coeffs calibration Series of measurements with predefined preamp parameters Sample&Holde mode

Parameters

|>p0.15|p0.805|

calcNorm	- true - calibration, false - measurements

Definition at line 728 of file analizer.cpp.

6.2.2.7 void doOnePulse (uint16_t *pulseWidth*)

Make one pulse to the LED with previously setted current and time of pulse

Definition at line 915 of file analizer.cpp.

6.2.2.8 void readADC (float & *value*)

Utility function for Sample&Holde mode of measurements make 16 samples and calc average,

Parameters

|>p0.10|>p0.15|p0.678|

`out` *value* buffer for reading voltage (mV)

Definition at line 823 of file analizer.cpp.

6.2.2.9 void readADCOneTime (uint16_t & *value*) `[inline]`

Utility function for inpulse mode of measurements

Parameters

|>p0.10|>p0.15|p0.678|

`out` *value* buffer for reading adc value (w/o convert to mV)

Definition at line 814 of file analizer.cpp.

6.2.2.10 void SerialClean ( )

Utility function for cleanserial port buffer

Definition at line 991 of file analizer.cpp.

6.2.2.11 void setC_R (float *val*)

Save to eeprom value of input resistanse of second OpAmp

Parameters

|>p0.15|p0.805|

val

Definition at line 910 of file analizer.cpp.

6.2.2.12 void setCurrent (uint8_t *channelN,* uint16_t *currValue*)

This function set currents for LED in channel channelN Control DAC Max value is 1000mA, with step 1mA After current will set, it send a message with currently setted values to serial port

Parameters

|>p0.15|p0.805|

channelN - Number of channel 1 or 2

currValue - DAC output voltage, max 1000mV it is equal 1000mA on LED

Definition at line 1044 of file analizer.cpp.

6.2.2.13 void setPreAmp (float *RWB1*, float *RWB2*)

This function set coeffs of OpAmp Control digital potentiometers ad5141 Max value is 100kOhm, with step 0.39k↩ Ohm After resistance will set, it send a message with currently setted values to serial port

Parameters

|>p0.15|p0.805|	
RWB1	1st cascade (current to voltage)
RWB2	2nd cascade (signal amplification)

Definition at line 1004 of file analizer.cpp.

6.2.2.14 void setPulseWidth (uint16_t *width*)

Set width of impulse in microseconds step is 1us, max 200us

Parameters

|>p0.15|p0.805|	
width	- time of pulse, max 200us

Definition at line 1089 of file analizer.cpp.

6.2.2.15 void setup ( )

Main initialization of MCU

Definition at line 27 of file analizer.cpp.

6.2.2.16 void shiftRegisterFirst ( )

Utility function for work with shift register Select first pair of led

Definition at line 970 of file analizer.cpp.

6.2.2.17 void shiftRegisterNext ( )

Utility function for work with shift register Select next led to ON

Definition at line 959 of file analizer.cpp.

6.2.2.18 void shiftRegisterReset ( )

Utility function for work with shift register Set all outputs to High level

Definition at line 934 of file analizer.cpp.

6.2.2.19 void writeConfigToUart ( )

Print all saved to eeprom data to serial port

Definition at line 843 of file analizer.cpp.

6.2.3 Variable Documentation

6.2.3.1 float EEMEM _c_R = 3.9

Definition at line 76 of file analizer.h.

6.2.3.2 float EEMEM _coefficients[NUM_OF_LED]

Definition at line 79 of file analizer.h.

6.2.3.3 uint8_t EEMEM _empty[20] = {0xF}

allocate eeprom variable

Definition at line 74 of file analizer.h.

6.2.3.4 etalon_t EEMEM _etalons[NUM_OF_ETALON]

Definition at line 78 of file analizer.h.

6.2.3.5 current_t EEMEM _pairsOfCurrent[NUM_OF_LED]

Definition at line 77 of file analizer.h.

6.2.3.6 uint16_t EEMEM _pulseWidth = 80

Definition at line 75 of file analizer.h.

6.2.3.7 volatile float coeffs[NUM_OF_LED]

Definition at line 84 of file analizer.h.

6.2.3.8 volatile current_t cur4AllLed[NUM_OF_LED]

Definition at line 83 of file analizer.h.

6.2.3.9 volatile bool oneTimes = false

Definition at line 85 of file analizer.h.

6.2.3.10 volatile bool pulseEnd =false

Definition at line 86 of file analizer.h.

6.3 /home/zvebabi/Documents/workspace/LEDMicrosensor_project0/analizer/↩ Release/analizer.d File Reference

6.4 /home/zvebabi/Documents/workspace/LEDMicrosensor_project0/analizer/↩ Release/SPI.d File Reference