

1. DESCRIPTION:

The **SENCS1Dx CONTROLLER** is a stand-alone hardware device designed specifically for the programming of SENCS1Dx current Sensors. It relies on the functionalities of STM32 Nucleo microcontroller board which manages both sensor communication and interaction with the user through simple interface. This microcontroller can also serve as a communication proxy between the computer and the Sensor, with sole purpose of coding and decoding signals in Manchester code format. The Controller features an LCD 20x4 display as its output device, while a set of five buttons serves as input devices ("UP", "DOWN", "LEFT", "RIGHT" and "SELECT"). These buttons are used for navigating through the user interface and initiating programming commands to the sensor.

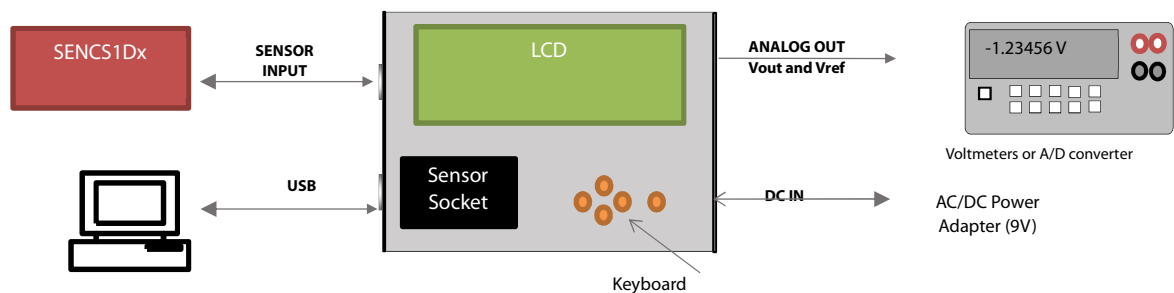


Figure 1: *Typical measurement setup with a SENCS1Dx Controller*



Figure 2: *Photo of the SENCS1Dx Controller*

2. ELECTRICAL SPECIFICATION:

Parameter	Value	Remarks
Supply Voltage	9V	None
Output voltages Vout & Vref	Single Ended	

3. MECHANICAL SPECIFICATION:

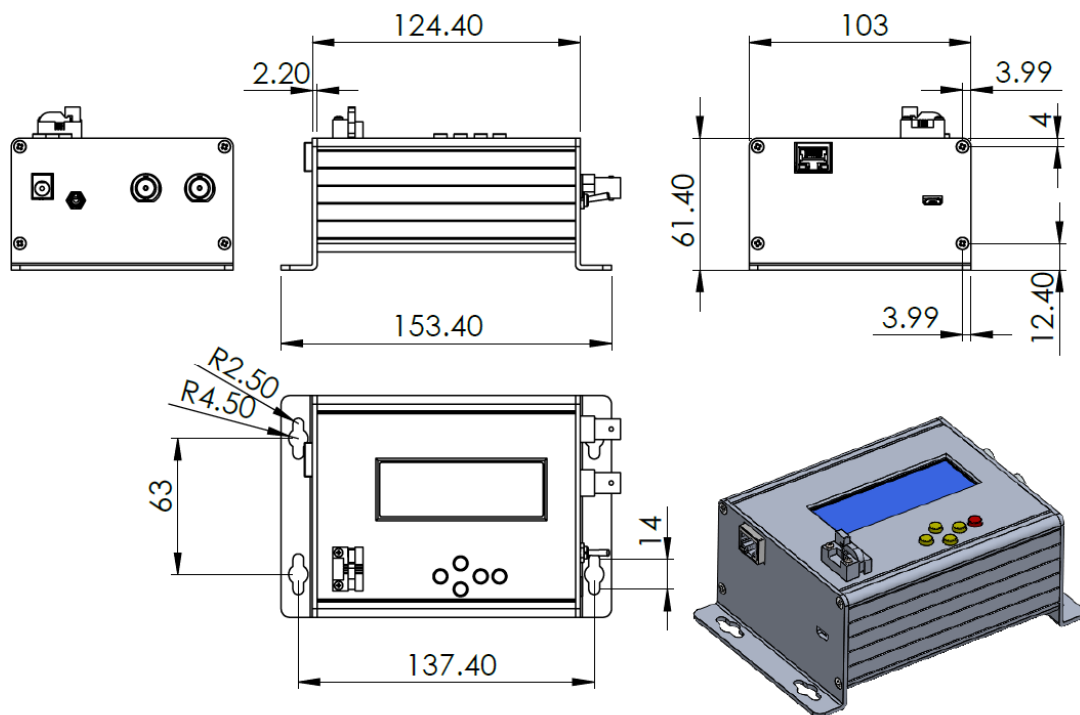


Figure 3: *Dimensions of the controller box (all dimensions are in millimeters (mm)).*

4. LCD OPERATION:

On device startup, a welcome screen is displayed for a couple of seconds, and it provides various details, including the software version. Afterward, the main menu of the controller is presented, offering the following options:

- Operation mode operations
- Register operations (read and write)
- OTP (One-Time-Programmable) memory operations (read and write)



Figure 4: *Photo of Controller during operation*

On the right side of the screen latest readings of Vout and Vref voltages are displayed. Also, above the main menu options the current operation mode is indicated with single letter sign ("N" – Normal mode, "R" – Register/Debug access mode, "P" – Programming mode).

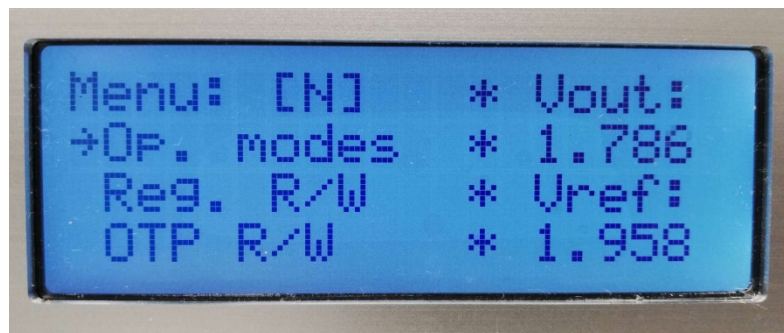


Figure 5: *AnyCS controller main menu*

Cursor represented by right arrow is used for option selection and can be navigated by using "UP" and "DOWN" buttons. For selection the option that cursor currently points to, "SELECT" button is used.

1.1 Operation modes menu

Every option from the main menu is associated with the corresponding submenu. By selecting the first option operation modes configuration menu will appear. This menu allows user to set sensor operation mode. Navigation through options remains the same as in the main menu with addition of back option ("LEFT" button) for returning to main menu.

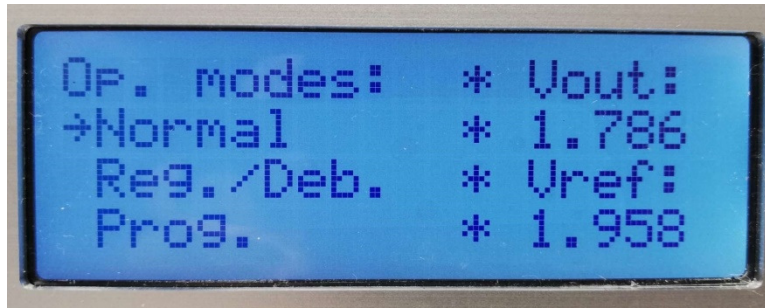


Figure 6: *Operation modes submenu*

When the user selects desired operation mode suitable mode change command is sent to sensor and confirmation screen appears to inform user that mode is changed. To exit confirmation screen "LEFT" button must be pressed which will return user to main menu.

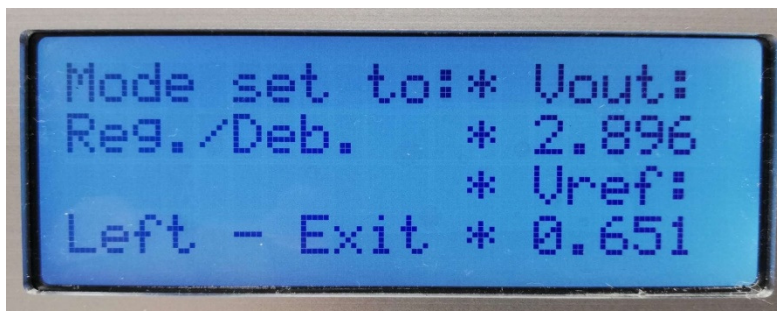


Figure 7: *Operation mode change confirmation screen*

1.2 Register operations menu

Selecting the "Reg. read/write" option from the main menu will open the register operation submenu. This submenu has greater depth compared to the previous one, which also makes it more complex. Firstly, user has to choose between read and write operation which will then lead to different but similar menus. It's worth noting that for register operations to work, the sensor must previously be set to "Register/Debug access mode".

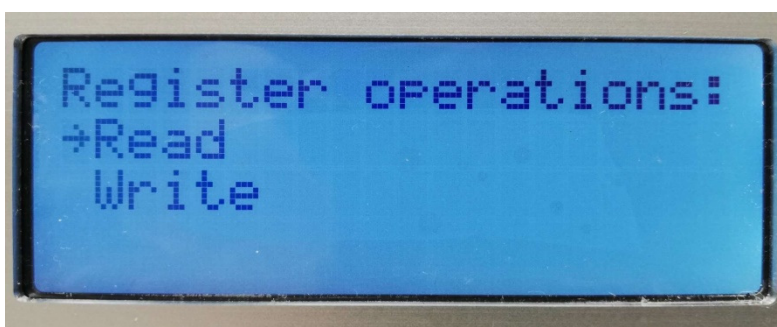


Figure 8: *Register operations submenu*

In the "Read Register" menu, the user only needs to select the register to be read. In contrast, the "Register Write" menu requires entering both the value to be written and the register. Navigation through registers is implemented in circular mode and accomplished by using "UP" and "DOWN" buttons. After navigation to desired register, reading operation is initiated by pressing "SELECT" button. If the reading is successful register value in hexadecimal will appear as "Read value", on the contrary if some error occurred during reading that error name will be shown instead of register value.

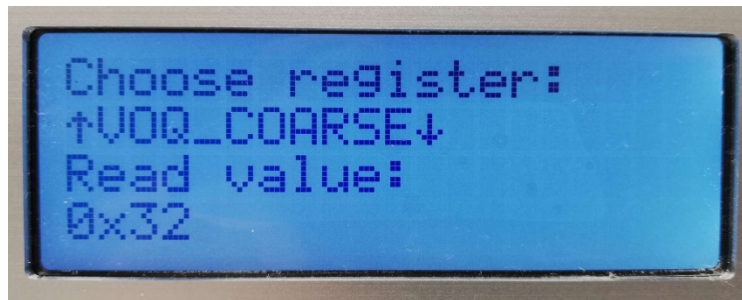


Figure 9: *Read register submenu in case of successful read*

In case of write operation, after selecting register, user have to also enter writing value. Register selection (confirmed by pressing "SELECT" button) is followed by the appearance of the blinking cursor at higher position digit (more significant digit) of writing value which is initially set to 0x00. Switching between higher and lower position digit is accomplished by "LEFT" and "RIGHT" buttons and changing digits values is accomplished through "UP" and "DOWN" buttons. Writing operation can be initiated in any moment during writing value editing by pressing "SELECT" button.

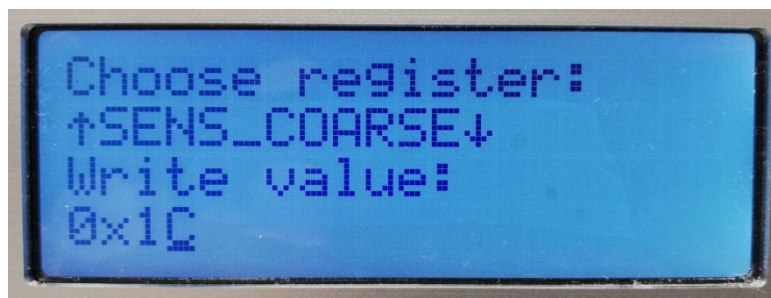


Figure 10: *Write register submenu during writing value edit*

When the writing command is executed new screen with writing result appears. In case of successful writing "REGISTER WRITING SUCCESSFUL!" message is displayed and in the case of error occurring the name of that error will be shown. This screen will last for one and the half seconds and after that user will be sent back to register operation submenu.

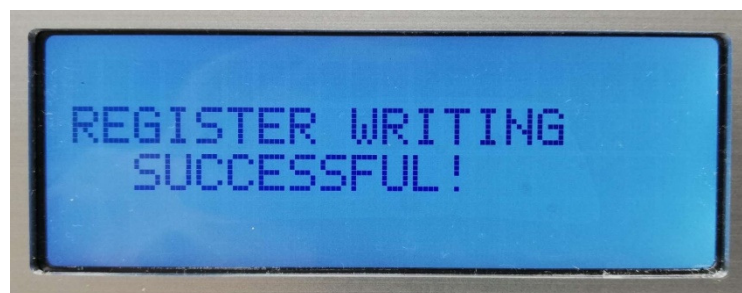


Figure 11: *Message for successfully performed register writing*

1.3 OTP programming menu

OTP operations submenu is almost identical to register operations submenu and offers two possibilities: reading and writing OTP memory. For the OTP operations to work sensor must be in "Programming mode". Read option opens up the screen similar to read register screen. User can then choose OTP page that he wants to read by using "UP" and "DOWN" buttons. User pages are represented with letter "U" followed by page number ("U0", "U1", "U2" and "U3") and factory page is represented with "F0" mark. Reading is initiated by pressing "SELECT" button.

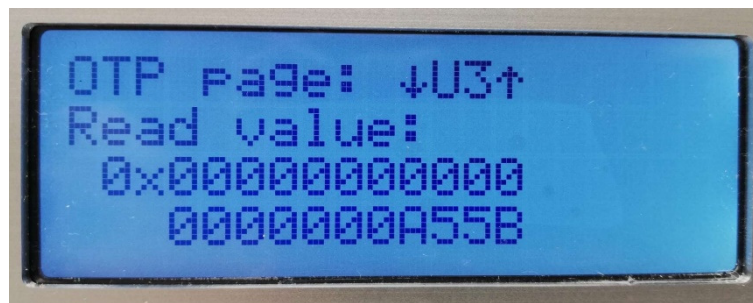


Figure 12: *OTP page read example*

In order to write to OTP memory user first has to choose OTP page and after that enter value that is going to be written. Writing value has 22 hex numbers (88 bits) of which first 18 represent 9 register values (factory or user registers depending on selected page) and can be modified freely, while last 4 represent validation code and checksum and can't be edited directly. Validation code has static value "A5" according to data-sheet and checksum value is dynamically calculated based on register locations values.

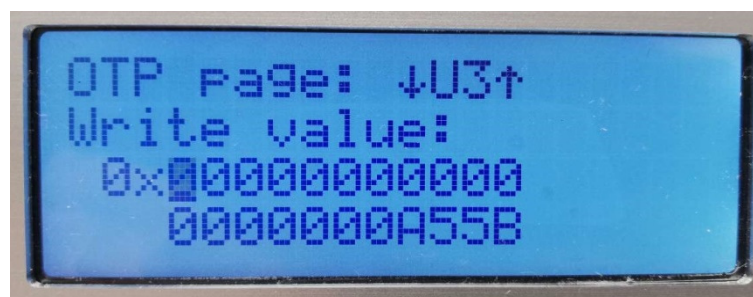


Figure 13: *OTP page write example*

It's important to note that the OTP writing process is irreversible, meaning that once written, it cannot be changed back to zero. Therefore, an additional confirmation screen is required, where the user is asked to confirm the writing operation.

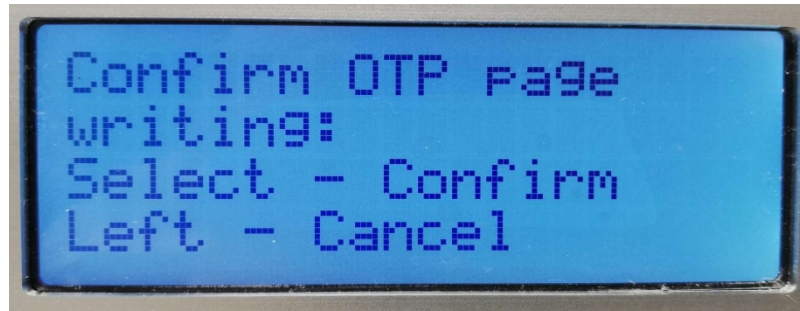


Figure 14: *OTP page write confirmation*

After the user's confirmation, the writing operation is initiated. Each logic one must be written separately in a command, which is why this operation may take more time than previously described operations. Once the writing is finished, the user is informed about the success rate, including how many writings were successful and how many failed.

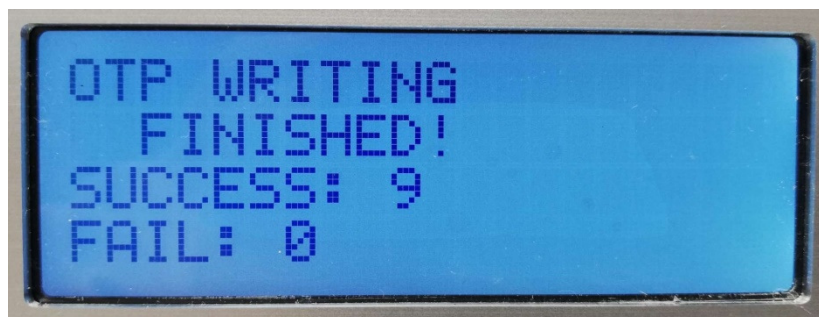


Figure 15: *OTP page write success rate*

5. PYTHON INTERFACE

As previously mentioned, the controller can serve as a Manchester code proxy between the computer and the sensor. This functionality opens up the opportunity to develop higher-level software that runs on the computer, which can be employed for sensor programming through controller using a USB cable. Expanding on this idea, additional Python software with graphical user interface has been implemented.

Aforementioned software is deployed in form of executable file and can be run without any additional installations. The primary requirement for this program to function is the presence of a serial interface. Therefore, the initial step when the program is launched is to establish serial communication by selecting one of the listed available ports.

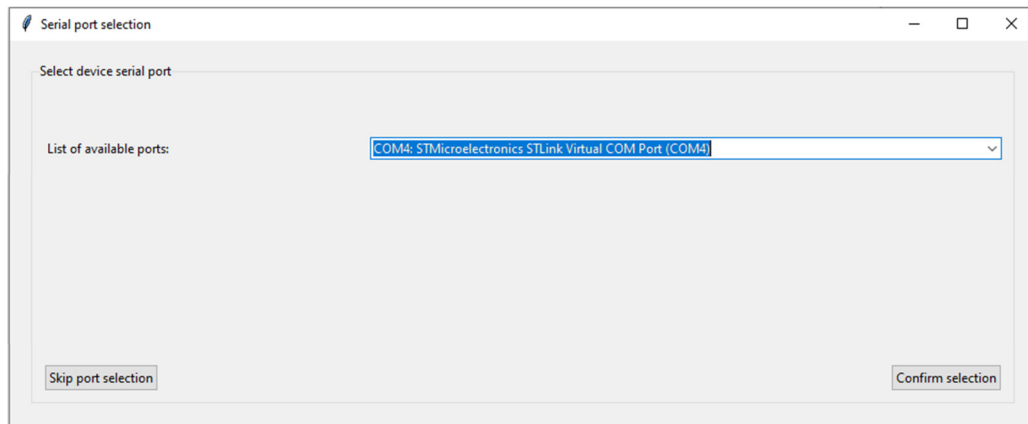


Figure 16: *Serial port selection*

After the serial port selection the application main window will appear. On the left side of this window is navigation menu with following options:

- Device configuration
- OTP memory programming
- Basic operations.

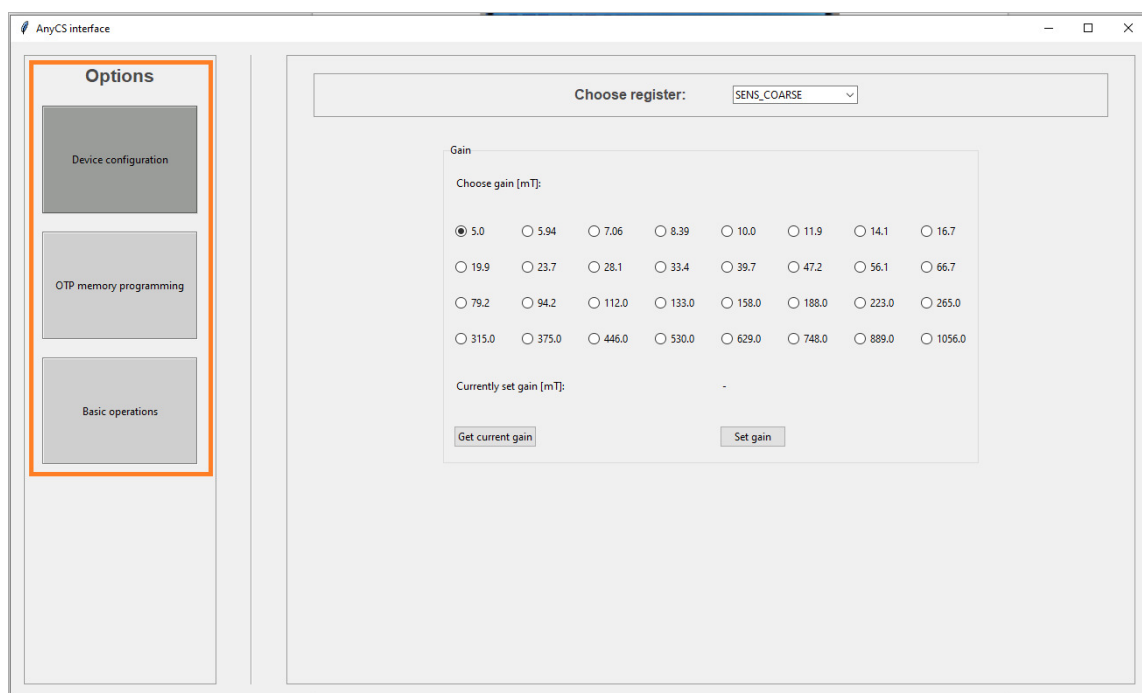


Figure 17: *Main window with highlighted navigation options*

By default, the “Device configuration” page is opened, allowing the user to select any register and modify settings through a user-friendly interface that doesn't require in-depth knowledge of sensor registers. For instance, if a user wants to switch the hall sensor axis from vertical to horizontal, they only need to choose the *GLOBAL_CFG* register, select the “Horizontal” axis within the “Hall sensor axis” label frame, and confirm their selection by clicking the “Set hall sensor axis” button.

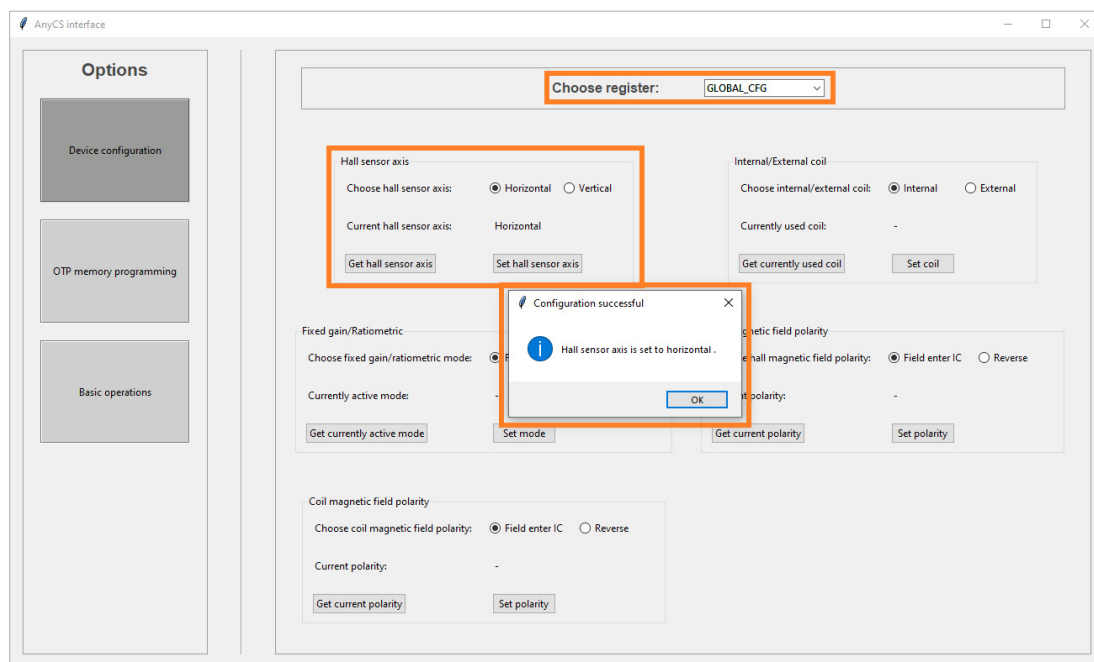


Figure 18: *Device configuration example*

User can also only read current settings by selecting *Get* operation instead of *Set*.

When selecting the second navigation option, the "OTP memory programming" page is accessed, which provides various OTP operations. This page is divided into two label frames. The first frame is dedicated to memory read, while the second one is designed for OTP page write.

The memory read frame is designed to be intuitive, displaying memory locations in a matrix format. Each column represents a memory page, and each row represents the same location across all pages. User pages are distinguishable from factory pages through different colors, with the specific colors chosen according to the datasheet. By clicking on "Read memory raw" button entire OTP memory (all five pages) will be read at once.

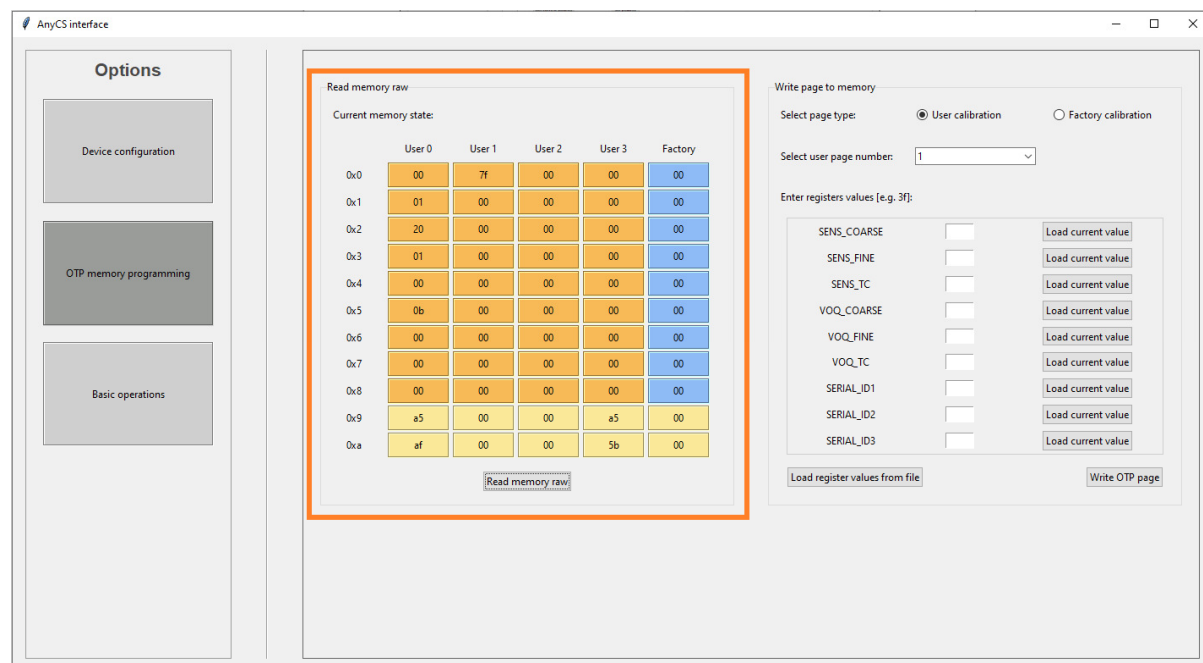


Figure 19: *OTP memory read*

In order to write OTP page user first has to choose page that is going to be written and after that enter register values (checksum will be calculated automatically). Register values can be entered directly (by typing in), loaded from register (current register value) or loaded from textual file.

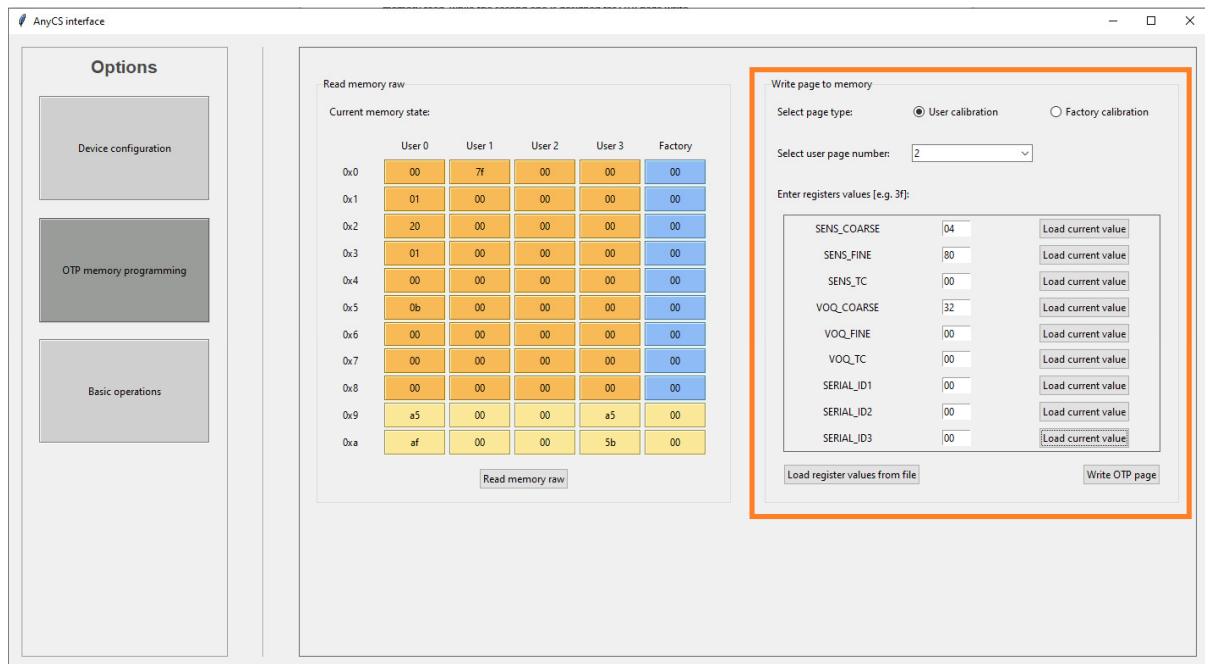


Figure 20: *OTP memory page write*

Last navigation option-“Basic operations” will open page that contains three basic operation label frames. First frame is “Single register operation” frame that enables user to both read from and write to single register. Next frame is in charge of operation mode change and it works circularly (*Normal mode* > *Register debug/access mode* > *Programming mode* > *Normal mode*). Operation mode value, displayed inside this frame will also be automatically updated if mode is changed from any other part of the software (e.g. selecting “Read memory raw” button will set operation mode to *Programming*). Last frame is used for multiple register operation, for example if user wants to read or write all register at once.

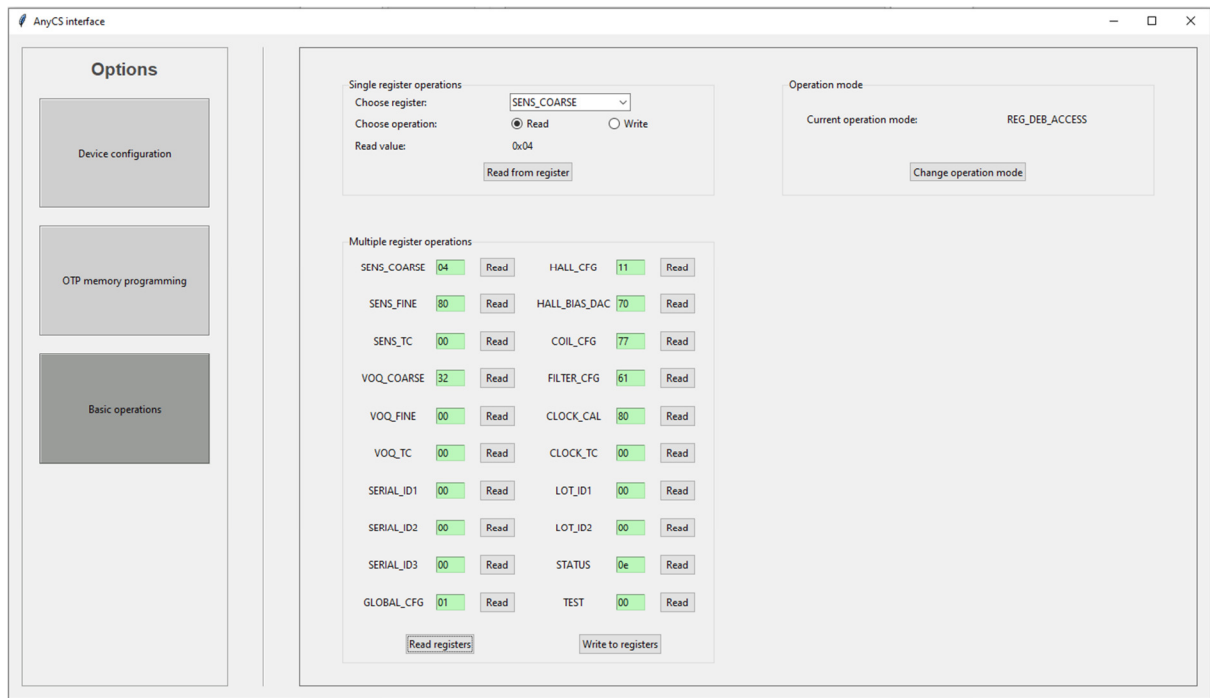


Figure 21: *Basic operations*

6. TABLE OF CONTENTS

1. DESCRIPTION:	1
2. ELECTRICAL SPECIFICATION:	2
3. MECHANICAL SPECIFICATION:.....	2
4. LCD OPERATION:	3
1.1 Operation modes menu.....	4
1.2 Register operations menu	4
1.3 OTP programming menu.....	6
5. PYTHON INTERFACE	7
6. Table of contents	12