Project Documentation - DARKBOX Environment Control

General Description:

The project is a system for managing and monitoring environmental parameters such as humidity, pressure, temperature, and light intensity, with configurable settings via a menu controlled through the serial port. The program uses various sensors (LM75, HTS221, TSL2571, LPS331), an RTC module, and EEPROM memory for storing and retrieving data.

Connected Devices:

1. **TSL2571** - Light intensity sensor.
2. **LPS331** - Pressure sensor.
3. **HTS221** - Humidity and temperature sensor.
4. **RTC_DS1307** - Real-time clock module.
5. **EEPROM** - External memory for storing configuration and sensor data.

Constants

LED Definitions

- `LEDR_PIN`: Pin for the red LED (default 13).
- `LEDG_PIN`: Pin for the green LED (default 12).

EEPROM Definitions

- `EEPROM_I2C_ADDRESS`: I2C address of the EEPROM module (0x50).
- `EEPROM_WRITE_DELAY`: Write delay for EEPROM operations (5 ms).
- `EEPROM_START_ADDRESS`: Start address for configuration storage (0x0001).
- `EEPROM_SENSOR_START_ADDRESS`: Start address for sensor data storage (0x0020).
- `EEPROM_SENSOR_END_ADDRESS`: End address for sensor data storage (4096).

Other Constants

- `NbCmds`: Number of available menu commands.

Key Global Variables

Configuration and Measurements

- `ConfigData config`: Structure holding critical system settings:
    - `upper_temp_value`: Upper critical temperature threshold (default 30.0°C).

    - `lower_temp_value`: Lower critical temperature threshold (default -5.0°C).
    - `critical_pressure`: Critical pressure threshold (default 1000 mbar).
    - `critical_humidity`: Critical humidity threshold (default 90%).
    - `alert`: Alert activation flag (default `false`).
- `float pressure, temperature, humidity, illuminance`: Variables storing the current measurement values.

LED Control

- `bool continous_display`: Flag enabling continuous display of measurement results.
- `bool LEDR_STATE, LEDG_STATE`: State of the LEDs (red and green).

Other

- `unsigned long prev_millis`: Variable for cyclic operation handling.
- `uint8_t CmdCode`: Code for the current user command.

Function Descriptions

EEPROM Functions

`clearEEPROM(uint16_t startAddress, uint16_t endAddress)`

- **Description**: Clears the EEPROM memory from the specified start address to the end address.
- **Parameters**:
    - `startAddress`: Starting address.
    - `endAddress`: Ending address.

`saveConfigToEEPROM()`

- **Description**: Saves the current system configuration to the EEPROM.
- **Parameters**: None.

`loadConfigFromEEPROM()`

- **Description**: Loads the system configuration from the EEPROM.
- **Parameters**: None.

`saveSensorDataToEEPROM(float temp, float humidity, float pressure, float illuminance, uint16_t startAddress, uint16_t endAddress)`

- **Description**: Saves sensor data to the EEPROM within the specified address range.
- **Parameters**:
    - `temp`: Temperature.

    - `humidity`: Humidity.
    - `pressure`: Pressure.
    - `illuminance`: Light intensity.
    - `startAddress`: Starting address.
    - `endAddress`: Ending address.

readSensorDataFromEEPROM(uint16_t startAddress)

- **Description**: Reads sensor data stored in the EEPROM from the specified start address.
- **Parameters**:
    - `startAddress`: Starting address.

isAddressEmpty(uint16_t address)

- **Description**: Checks if the given EEPROM address is empty.
- **Parameters**:
    - `address`: Address in the EEPROM.

Configuration Functions

do_set_h()

- **Description**: Sets the critical humidity value based on user input.
- **Parameters**: None.

do_set_p()

- **Description**: Sets the critical pressure value based on user input.
- **Parameters**: None.

do_set_tl()

- **Description**: Sets the lower critical temperature threshold.
- **Parameters**: None.

do_set_th()

- **Description**: Sets the upper critical temperature threshold.
- **Parameters**: None.

do_set_date()

- **Description**: Allows the user to set the current date and time.
- **Parameters**: None.

Display Functions

do_settings()

- **Description**: Displays the current critical settings (temperature, humidity, pressure).
- **Parameters**: None.

do_display()

- **Description**: Displays the current measurement data (temperature, humidity, pressure, light intensity).
- **Parameters**: None.

LED Handling Functions

temp_check()

- **Description**: Checks if the temperature settings are valid (i.e., the lower value does not exceed the upper value).
- **Parameters**: None.

Helper Functions

isValidDate(int year, int month, int day, int hours, int minutes, int seconds)

- **Description**: Validates the entered date.
- **Parameters**:
  - `year`, `month`, `day`, `hours`, `minutes`, `seconds`: Components of the date to be validated.

resetFunc()

- **Description**: Resets the microcontroller.
- **Parameters**: None.

Main Functions

setup()

- **Description**: Initializes the system, including sensors, the RTC module, EEPROM, and the menu.
- **Parameters**: None.

loop()

- **Description**: The main program loop, responsible for handling sensors, updating LED states, and processing menu commands.
- **Parameters**: None.