Documentation for sensorModelo06 Project

Overview

The sensorModelo06 project is an Arduino-based system designed for environmental monitoring and data collection. It integrates multiple sensors, GPS, GSM/LTE communication, and AES encryption to collect, process, and securely transmit data. The system is designed to operate efficiently with low power consumption, utilizing deep sleep modes to conserve energy.

Key Features

1. **Sensor Integration**:
    - Collects data from various environmental sensors (e.g., temperature, humidity, light, soil properties).
2. **GPS Module**:
    - Retrieves and stores latitude and longitude for geolocation.
3. **GSM/LTE Communication**:
    - Sends collected data to a remote server using TCP.
4. **AES Encryption**:
    - Encrypts data before transmission for secure communication.
5. **Low Power Mode**:
    - Utilizes deep sleep to conserve energy during idle periods.

File Structure

1. sensorModelo06.ino:
    - Main entry point for the Arduino program.
    - Initializes components and manages the main loop.
2. gpssen.cpp **/** gpssen.h:
    - Handles GPS data collection, parsing, and storage.
3. sensores.cpp **/** sensores.h:
    - Manages sensor initialization, data collection, and debugging.
4. gsmlte.cpp **/** gsmlte.h:

    - Manages GSM/LTE communication, including TCP connections and modem control.
5. aesenc.cpp / aesenc.h:
    - Provides AES encryption and decryption for secure data transmission.
6. timeData.cpp / timeData.h:
    - Handles time-related operations, including RTC synchronization and epoch time management.
7. sleepSensor.cpp / sleepSensor.h:
    - Manages deep sleep mode and wake-up handling.
8. type_def.h:
    - Defines the sensordata_type structure for storing sensor data.

Global Constants

- SEND_DATA_INTERVAL:
    - Interval (in milliseconds) for sending data.
- MAX_RETRIES:
    - Maximum number of retries for sending data before entering sleep mode.

Global Variables

- sensordata:
    - Stores sensor data in a structured format.
- uuid:
    - Unique identifier for the device.
- epoc:
    - Epoch time as a string.
- latitud / longitud:
    - Latitude and longitude as strings.
- rssi:
    - Signal strength (e.g., "-70dBm").

Main Components

1. Sensors

- **Initialization**:
  - Sensors are initialized in setupSensores() with retries for robustness.
- **Data Collection**:
  - Sensor data is read and stored in the sensordata structure using readSensorData().
- **Debugging**:
  - Sensor data can be printed to the serial monitor using debugData().

2. GPS

- **Initialization**:
  - GPS module is initialized in setupGps().
- **Data Handling**:
  - GPS data is read, validated, and stored using readGps() and parseGNRMC().

3. GSM/LTE Communication

- **Initialization**:
  - GSM/LTE module is initialized in inicioGsmLte().
- **Data Transmission**:
  - Data is sent to a remote server using tcpOpen(), tcpSendData(), and tcpClose().

4. AES Encryption

- **Encryption**:
  - Data is encrypted using encrypt() before transmission.
- **Decryption**:
  - Encrypted data can be decrypted using decrypt().

5. Time Management

- **RTC Synchronization**:
  - Internal RTC is synchronized with an external RTC using setFechaRtc().
- **Epoch Time**:
  - Epoch time is updated and stored using epoch().

6. Sleep Mode

- **Deep Sleep**:
    - The system enters deep sleep mode using sleepIOT() to conserve energy.
- **Wake-Up Handling**:
    - Wake-up reasons are printed and handled using printWakeupReason().

Workflow

Setup Phase

1. Initialize serial communication.
2. Print wake-up reason.
3. Initialize time, sensors, GPS, and GSM/LTE modules.

Main Loop

1. Check if the data send interval has elapsed.
2. Read sensor data and debug it.
3. Encrypt the sensor data.
4. Send the encrypted data to the server.
5. Enter sleep mode after successful data transmission or retries.

Key Functions

agro.ino

- setup():
    - Initializes all components.
- loop():
    - Manages the main workflow, including data collection, encryption, and transmission.
- check_interval():
    - Checks if the data send interval has elapsed.
- convertStructToString():
    - Converts the sensordata structure to a concatenated string.
- envioData():

    - Sends data over GSM/LTE with retries.

gpssen.cpp

- setupGps():
    - Initializes the GPS module.
- runGpsSen():
    - FreeRTOS task for continuously reading GPS data.
- readGps():
    - Reads GPS data from the serial port.
- parseGNRMC():
    - Parses GPS data to extract latitude and longitude.

sensores.cpp

- setupSensores():
    - Initializes sensors with retries.
- readSensorData():
    - Reads data from all connected sensors.
- debugData():
    - Prints sensor data for debugging.

gsmlte.cpp

- inicioGsmLte():
    - Initializes the GSM/LTE module.
- startLTE():
    - Starts the LTE connection.
- tcpSendData():
    - Sends data over TCP with retries.

aesenc.cpp

- encrypt():
    - Encrypts plaintext using AES and encodes it in Base64.
- decrypt():
    - Decrypts Base64-encoded AES-encrypted data.

timeData.cpp

- setupTimeData():
    - Initializes time-related data.
- setFechaRtc():
    - Synchronizes the internal RTC with the external RTC.
- epoch():
    - Updates the epoch time.

sleepSensor.cpp

- sleepIOT():
    - Puts the system into deep sleep mode.
- printWakeupReason():
    - Prints the reason for waking up from sleep.

Example Output

- **Serial Monitor**

```
ESP-ROM:esp32s3-20210327
Build:Mar 27 2021
rst:0xc (RTC_SW_CPU_RST),boot:0xb (SPI_FAST_FLASH_BOOT)
Saved PC:0x4037a22a
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fce2820,len:0x118c
load:0x403c8700,len:0x4
load:0x403c8704,len:0xc20
load:0x403cb700,len:0x30e0
entry 0x403c88b8
Wakeup was not causeESP-ROM:esp32s3-20210327
Build:Mar 27 2021
rst:0x1 (POWERON),boot:0xb (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fce2820,len:0x118c
load:0x403c8700,len:0x4
load:0x403c8704,len:0xc20
load:0x403cb700,len:0x30e0
entry 0x403c88b8
Wakeup was not caused by deep sleep: 0
1747403304
Task GPS running on core 0
20.6447449
0.0000000
======== INIT LTE ========
Modem: R1951.07
Connected to network.
-----Debug----
uuid: 89883030000096467599
epoc: 1747403304
temperatura_ambiente: 31.74
humedad_relativa : 22.61
luz: 12.50
radiación: 0.00
presion_barometrica: 0.00
co2: 0.00
humedad_suelo: 3.82
temperatura_suelo: 27.89
conductividad_suelo: 0.01
ph_suelo: 0.00
nitrogeno: 0.00
fosforo: 10.10
potasio: 0.00
viento: 0.00
latitud: 20.6447449
logitud: 0.0000000
battery voltage: 3.55
rssi: 10
---------
String constructed to send:
$,89883030000096467599,1747403304,31.74,22.61,12.50,0.00,0.00,0.00,3.82,27.89,0.01,0.00,0.00,10.10,0.00,0.00,20.6447449,0.0000000,3.55,10,#

Data sent successfully on attempt 1

Setup ESP32 to sleep for every 600 seconds
Going to sleep now
```

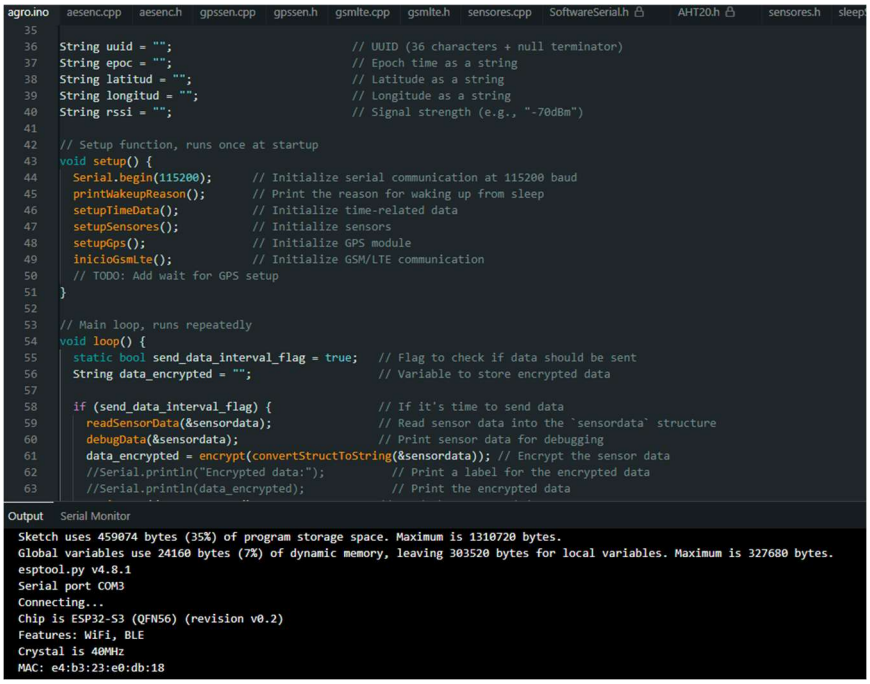
Key Notes

- **Error Handling**:
    - Retries are implemented for sensor initialization and data transmission.
    - The system enters sleep mode after maximum retries.
- **Low Power Design**:
    - Deep sleep mode is used to conserve energy during idle periods.
- **Modular Design**:
    - Functions are modularized for better readability and maintainability.

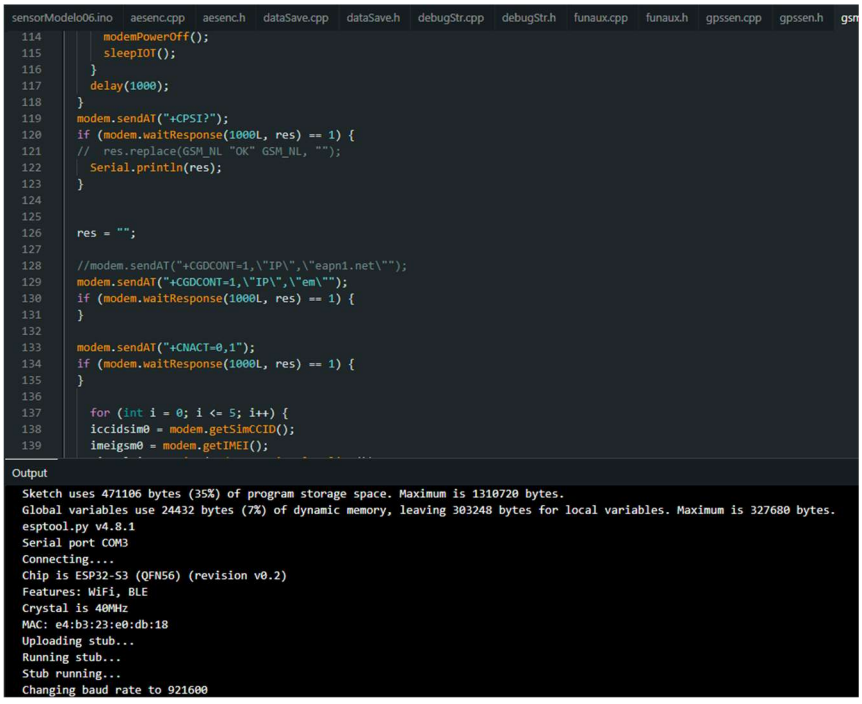
Server side data store

Comparative power measures, optimized firmware vs. original firmware

Memory

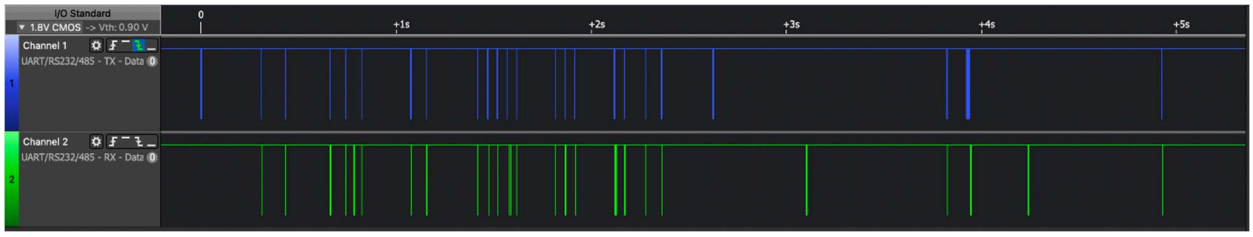


Optimized firmware.

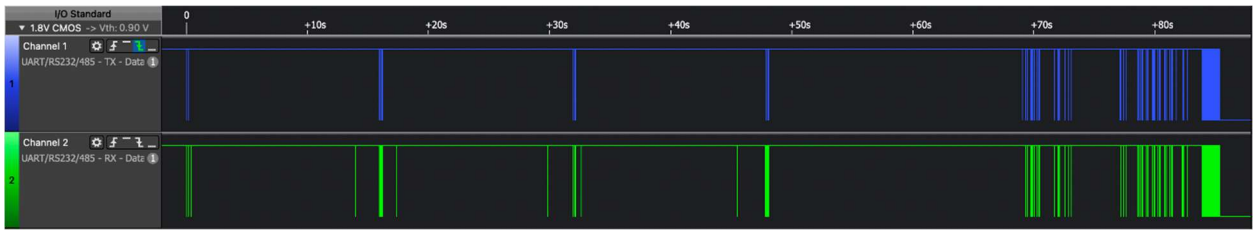


Original firmware.

Modem time used from boot to end send message.

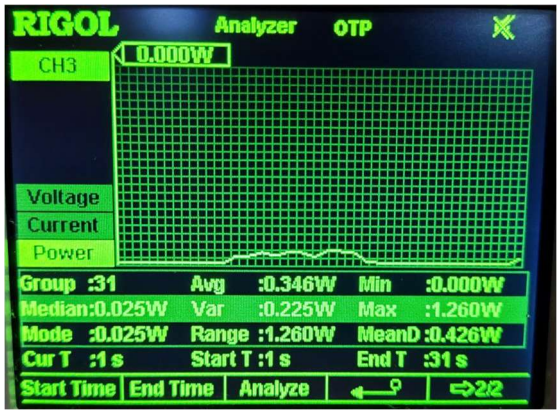


Optimized firmware.

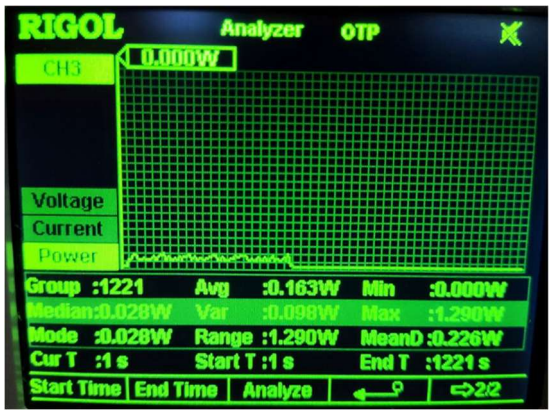


Original firmware.

Power battery drain



Optimized firmware.



Original firmware.