TÉCNICO LISBOA

DEEC
DEPARTAMENTO DE ENGEN
ELECTROTÉCNICA E DE COM
TÉCNICO LISBOA

Microchip Sensor Controller

Programmable Electronic Systems

MEE 2024

Group 1
Manuel Passadouro 80840
Miguel Simões 99162

TÉCNICO LISBOA

Objective

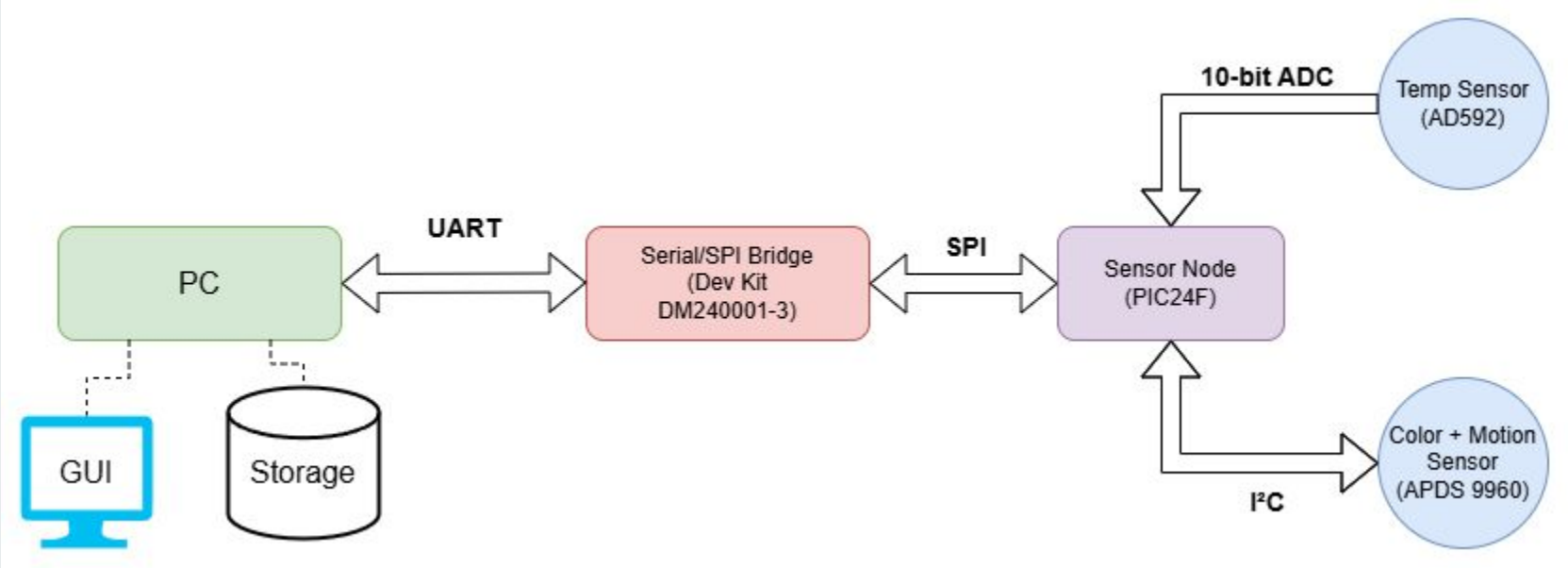
- Develop a system capable of acquiring and displaying sensor data about its surroundings;
- Relevant data: Temperature, Ambient Light, Object Proximity and Color;
- Possible applications: Motion based control (digital painting), Assembly line monitoring (Car production line, distribution centers).

TÉCNICO LISBOA

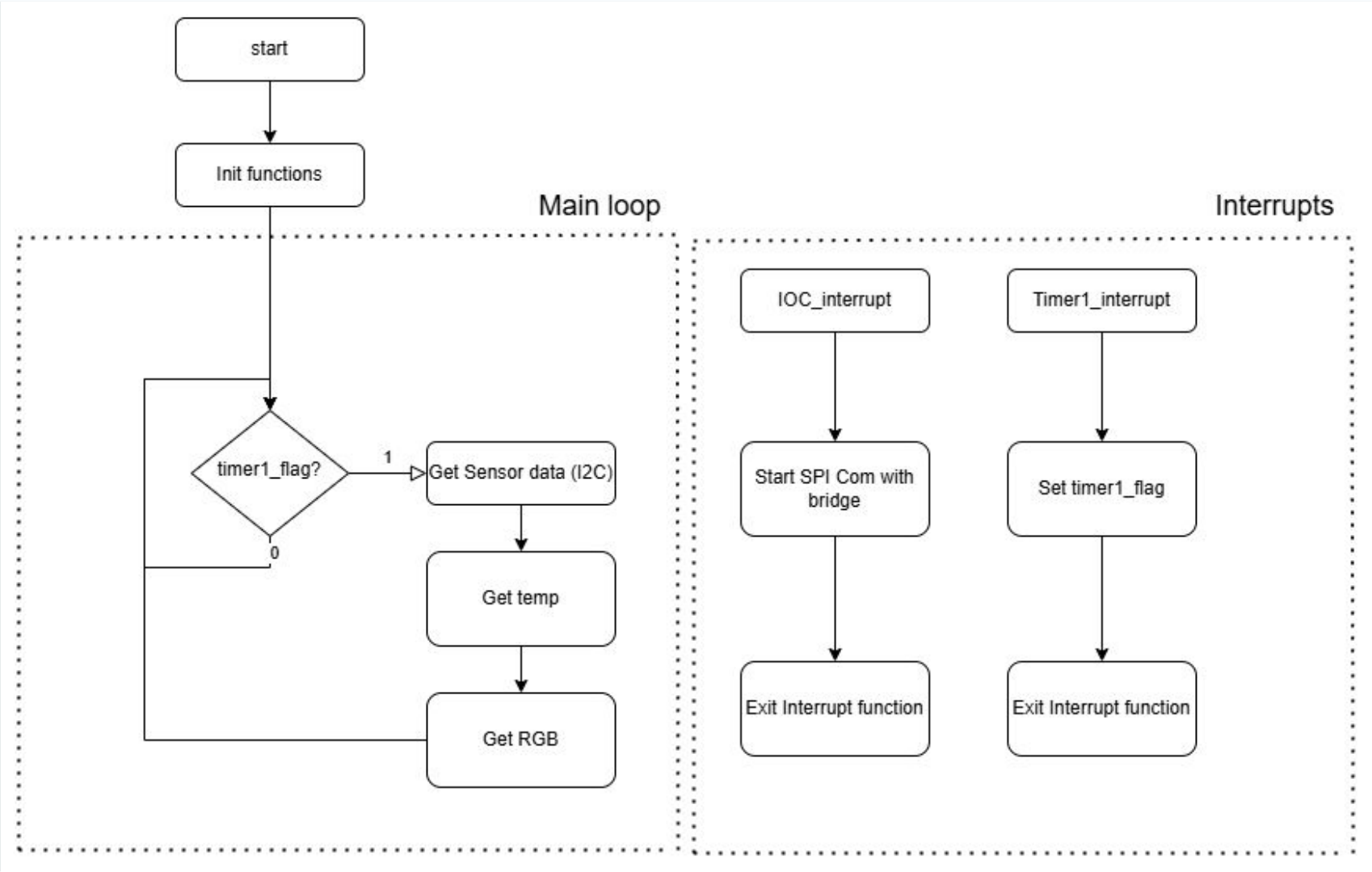
Features

- Read ambient temperature using a AD592 sensor;
- Measure Prox., ALS and Color using an APDS 9960 sensor;
- Send sensor data from sensor node to the PC, via the Serial/SPI Bridge;
- Display Sensor data on a GUI;
- Store timestamped sensor data on the PC in a machine readable format (i.e. csv);
- Enter a low power mode when not acquiring or sending data.

System Overview



Sensor Node Flowchart



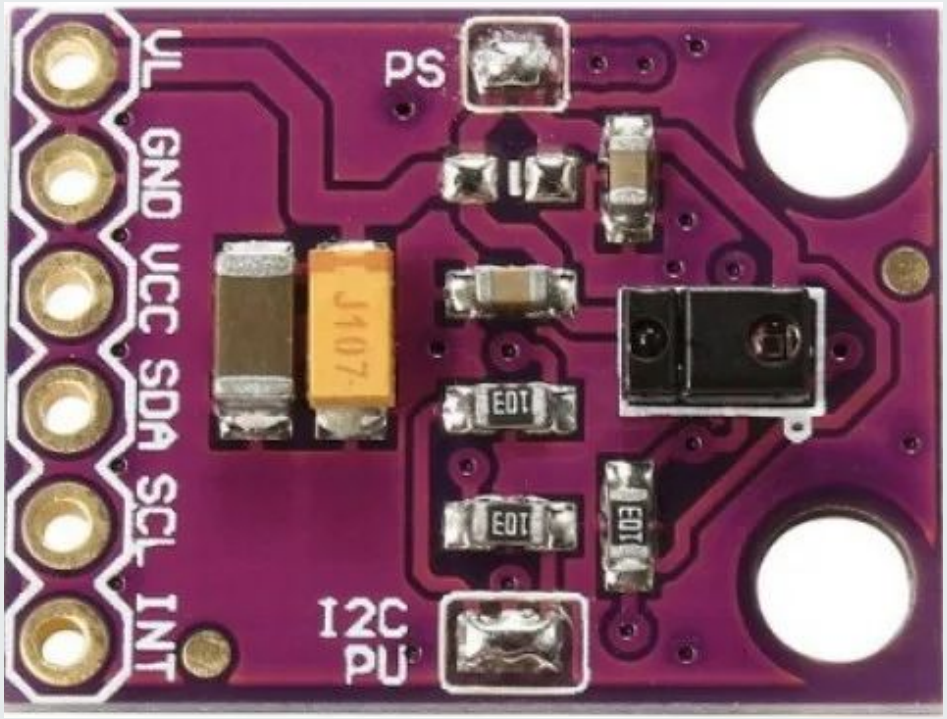
APDS 9960

- Proximity Sensor

Writes proximity register address, reads data, and stores it in prox_data.

- Gesture Sensor
- RGB Sensor

Similar write/read steps are used to get high and low bytes for each color channel, storing them in respective pointers (_get_red, _get_green, _get_blue).



TÉCNICO LISBOA

I2C

- Sets 7-bit address mode and a 400kHz communication speed;
- Configures the appropriate pins (RB8 and RB9) as inputs and sets them to open-drain;
- Generates a start and stop condition on the I2C bus until the sequence is completed;
- Sends a byte of data to the I2C bus and then reads a byte from the I2C bus. It first enables the receive mode, waits for data to be available in the receive buffer, and then retrieves the data.

Interrupt On Change (IOC)

- Manages interrupts triggered by changes on configured I/O pins.
- Clears the interrupt flag and calls spi_slave_handle() for SPI communication;
- Activates IoC for all pins, enables falling edge detection for pin B14;
- Clears interrupt flags, sets priority to 4, and enables IoC interrupts;
- Enables global interrupts to allow system-wide interrupt handling.

Timer

Timer1 reaches 1s of period using:

- Biggest prescaler, 1:256
- LPRC oscillator low frequency, 31kHz

To increase the step for each count (longer max period);

Counts within the 16 bit operation range:

- PR1= 120 counts

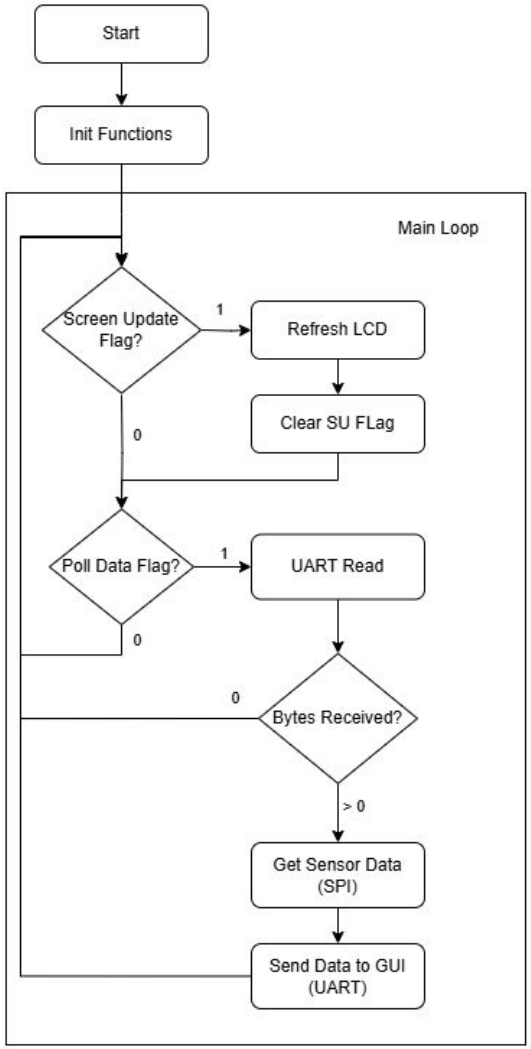
Timer1 Interrupts with priority 3 (lower than IoC)

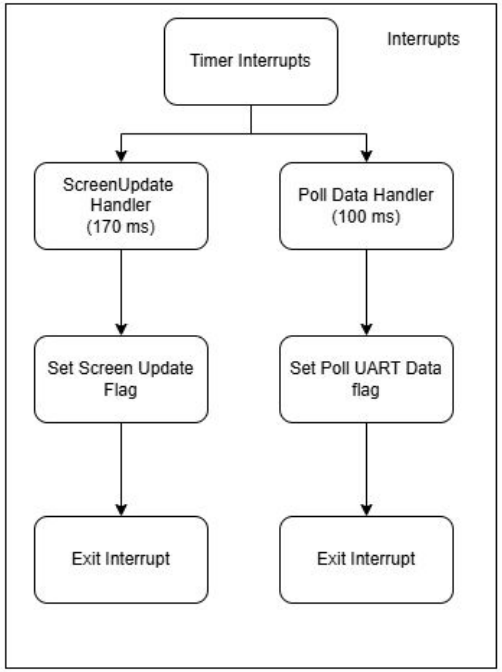
TÉCNICO LISBOA

ADC

- Configures pin RB3 as an analog input for sensor data.
- Sets internal counter sampling with a sample time of 15Tad.
- Selects RB3 as the input channel and disables scanning of additional channels.
- Clears the ADC interrupt flag and starts auto-sampling, Reads two samples, averages them, and returns the result as two 8-bit bytes (high and low).
- Converts the ADC voltage value to a temperature in Celsius by translating the 10-bit ADC value to Kelvin and adjusting to Celsius.

Bridge Flowchart





TÉCNICO LISBOA

Serial Peripheral Interface

- Configuration: 8-bit mode, mid-bit sampling, 100 kHz baud rate. Sets SCK idle low, data change on falling edge, and master mode.
- Pin and Peripheral Setup: Disables analog functions on GPIO pins, assigns pins for SCK, MISO, MOSI, and CS using PPS (Peripheral Pin Select).
- Enable SPI: Sets CS high (slave enabled) and activates SPI.

Serial Peripheral Interface

- **Start Communication:** Master (Bridge) sets CS low, triggers Slave (Sensor Node) IoC interrupt;
- **Send and Receive Byte:** After a delay, Master writes a byte to MOSI line to waiting Slave. Simultaneously the slave writes a reply in the MOSI line.

Receive Data Based on Command:

- The first byte sent by the master will be a command, this will tell the Slave which sensor data to write on the output buffer;
- Master sends Dummy bytes to loop through the Slave data, which will then be sent via UART;
- When there is no more data to request from the slave, Master set CS to 1, end communication.

TÉCNICO LISBOA

UART

UART1 Initialization :

- **Basic Configuration:** 1 stop bit, no parity, 8 data bits, and disables auto-baud.
- **Baud Rate:** Configures baud rate (9600) using U1BRG.
- **Transmit Enable:** Enables UART transmission.

Pin Configuration (PPS):

- Mapping TX/RX Pins: Configures U1RX on pin 49 and U1TX on pin 50, with analog functionality disabled.

TÉCNICO LISBOA

UART

- **UART Buffer:** Fixed length buffer of type char (10 Bytes);
- **Send Data:** Loops through a buffer and sends data byte-by-byte via UART, inserting a short delay between each byte to avoid overflow.
- **Non-Blocking Polling:** Reads available data from UART into a buffer up to its size limit, returning the number of bytes read.

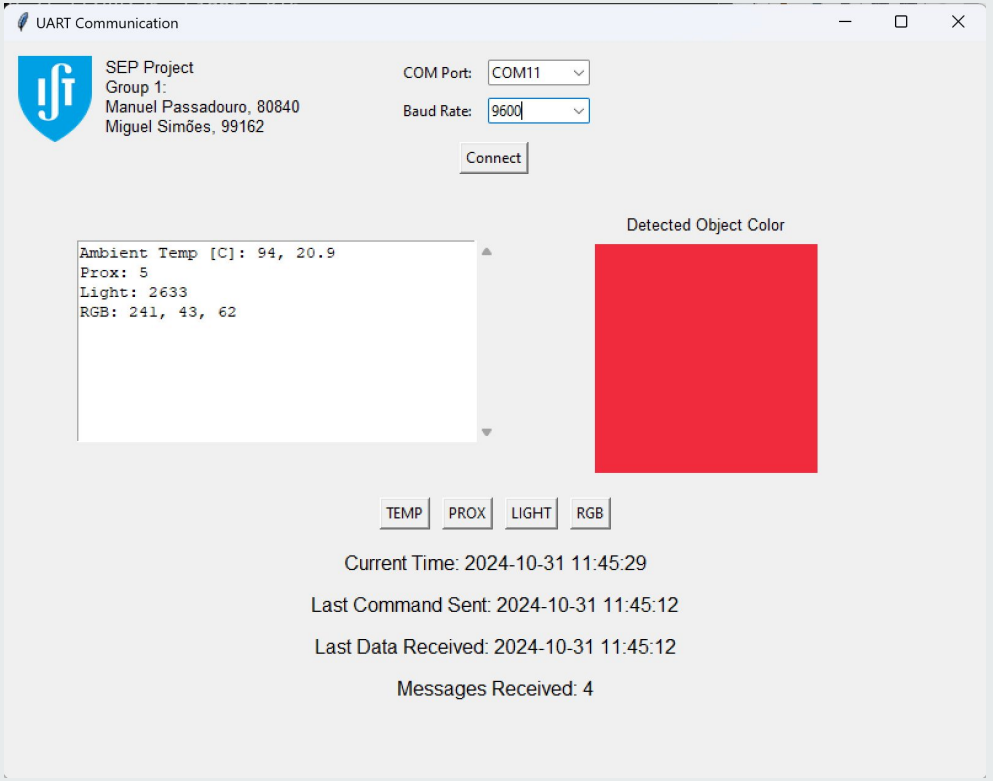
TÉCNICO LISBOA

Graphical User Interface

- **Setup:** Initializes a CSV file for logging, GUI elements, and variables for UART settings.
- **UART Connection:** Allows users to select a COM port and baud rate, then establishes the connection.
- **Data Reading:** Continuously reads incoming UART data, parses sensor values, and displays them in a text area.
- **Command Buttons:** Provides buttons to send specific commands (TEMP, PROX, LIGHT, RGB) to the UART device.

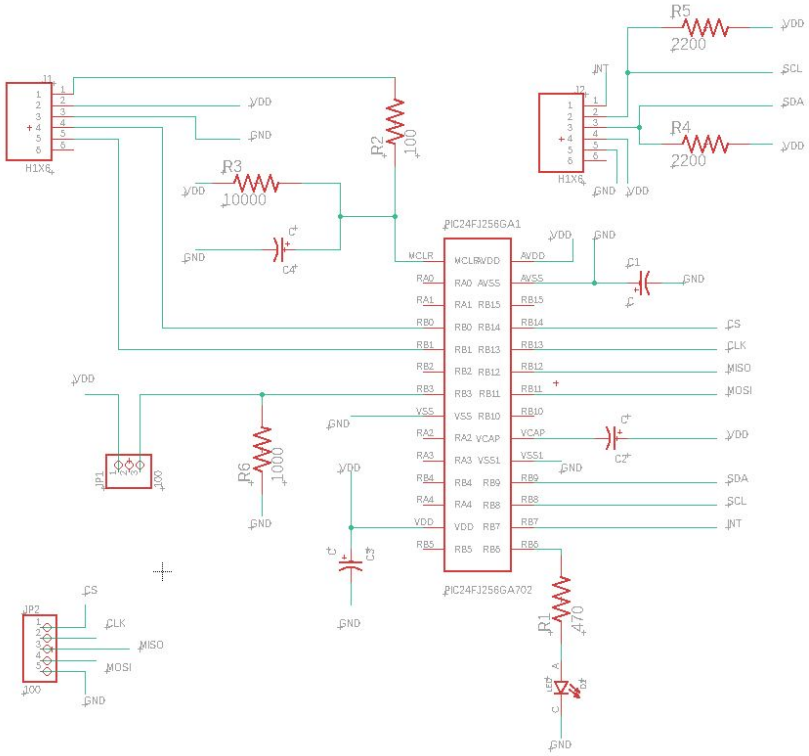
Graphical User Interface

- **Live Color Display:** Updates a color square based on RGB data received, representing detected object colors.
- **Status Display:** Shows current time, last command sent, last data received, and total messages.
- **Logging:** Logs each data entry with a timestamp in the CSV file.

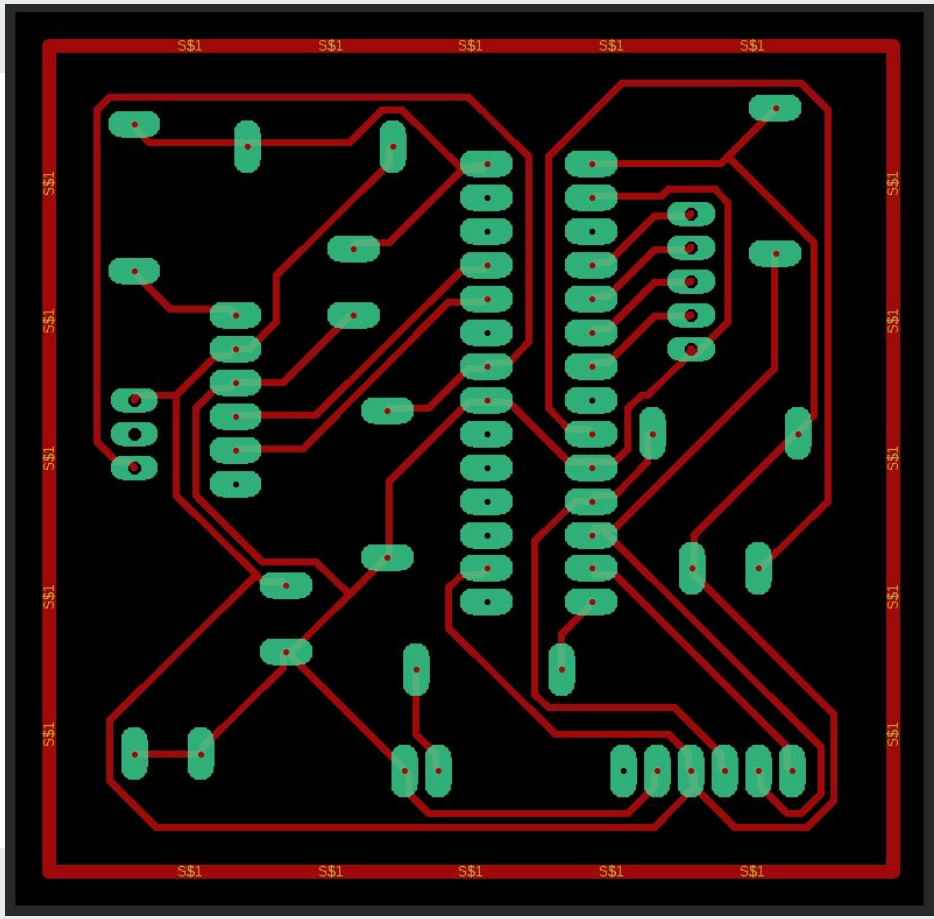


Printed Circuit Board

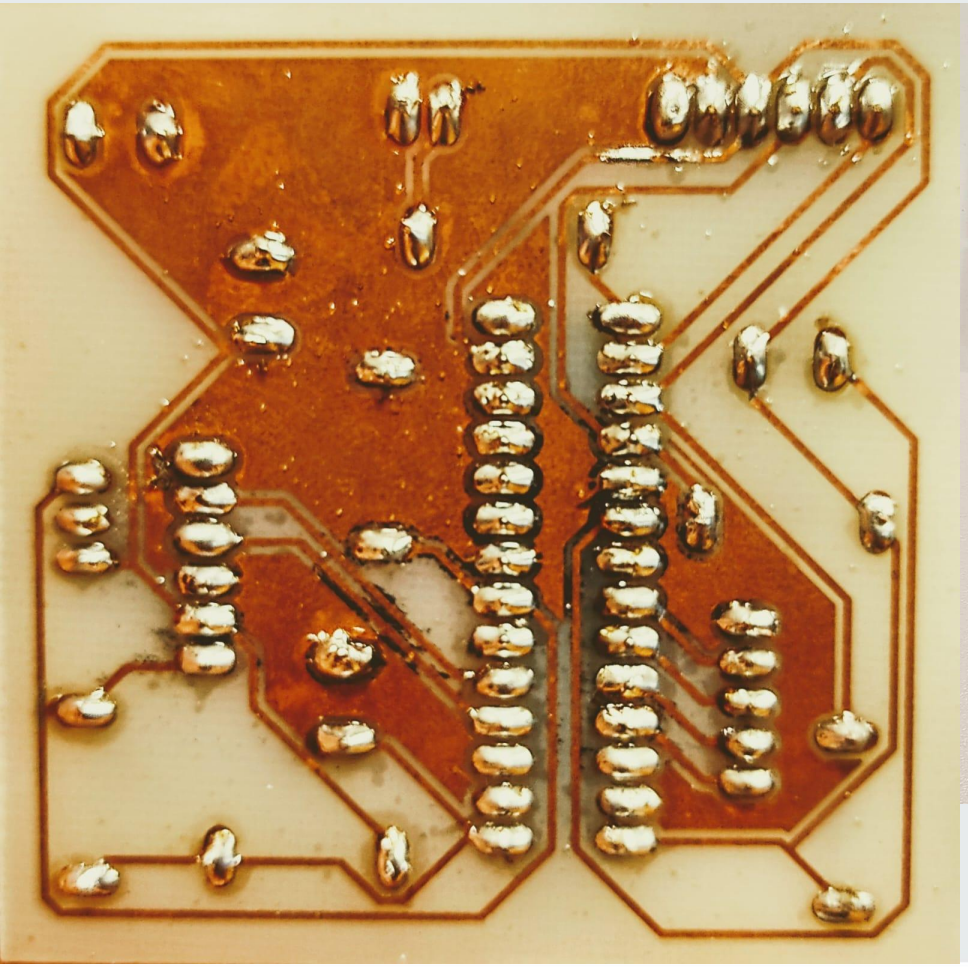
Schematic

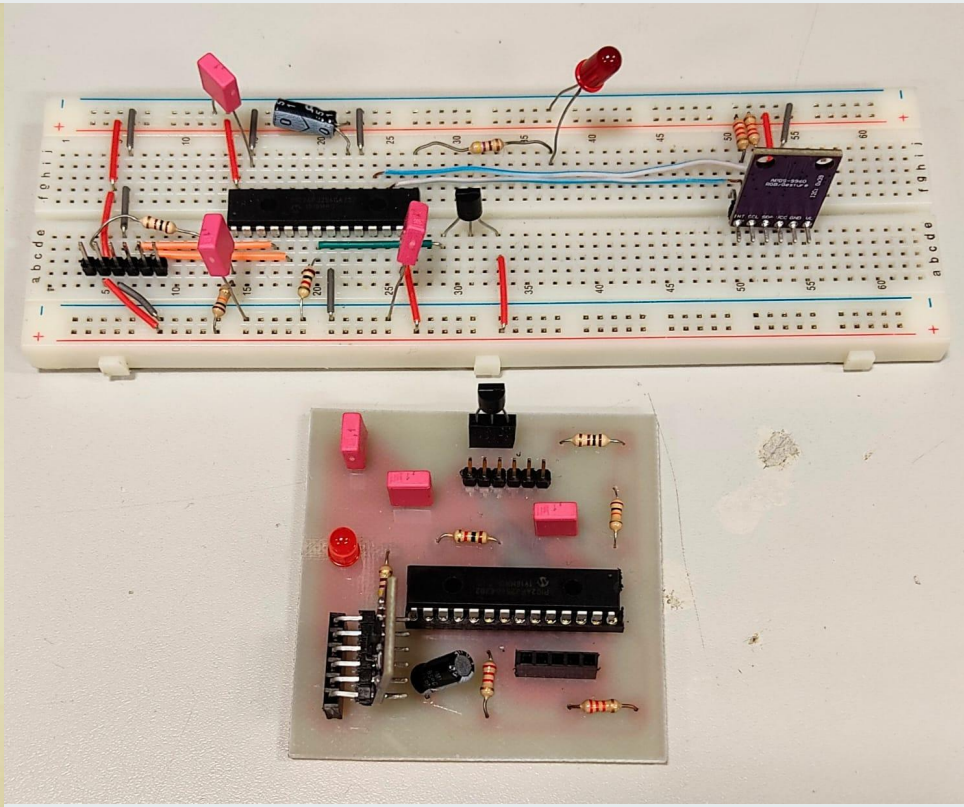


Layout



Printed Circuit Board





Conclusion

All code and more detailed information is available at:
https://github.com/manuel-passadouro/SEP_Project

Thank you for your attention!