E-P.I.P.

Environmental and Personal Information Processor

Team Member, Contact	Contributions
Alberto Rosas, Project Leader arosas28@horizon.csueastbay.edu	MCU, Micro SD Card
Cody Hum chum@horizon.csueastbay.edu	AIR, TEMP, I/O
Dang Tran dtran134@horizon.csueastbay.edu	GPS, Power Supply
Noel Anilao nanilao2@horizon.csueastbay.edu	Vitals, LCD

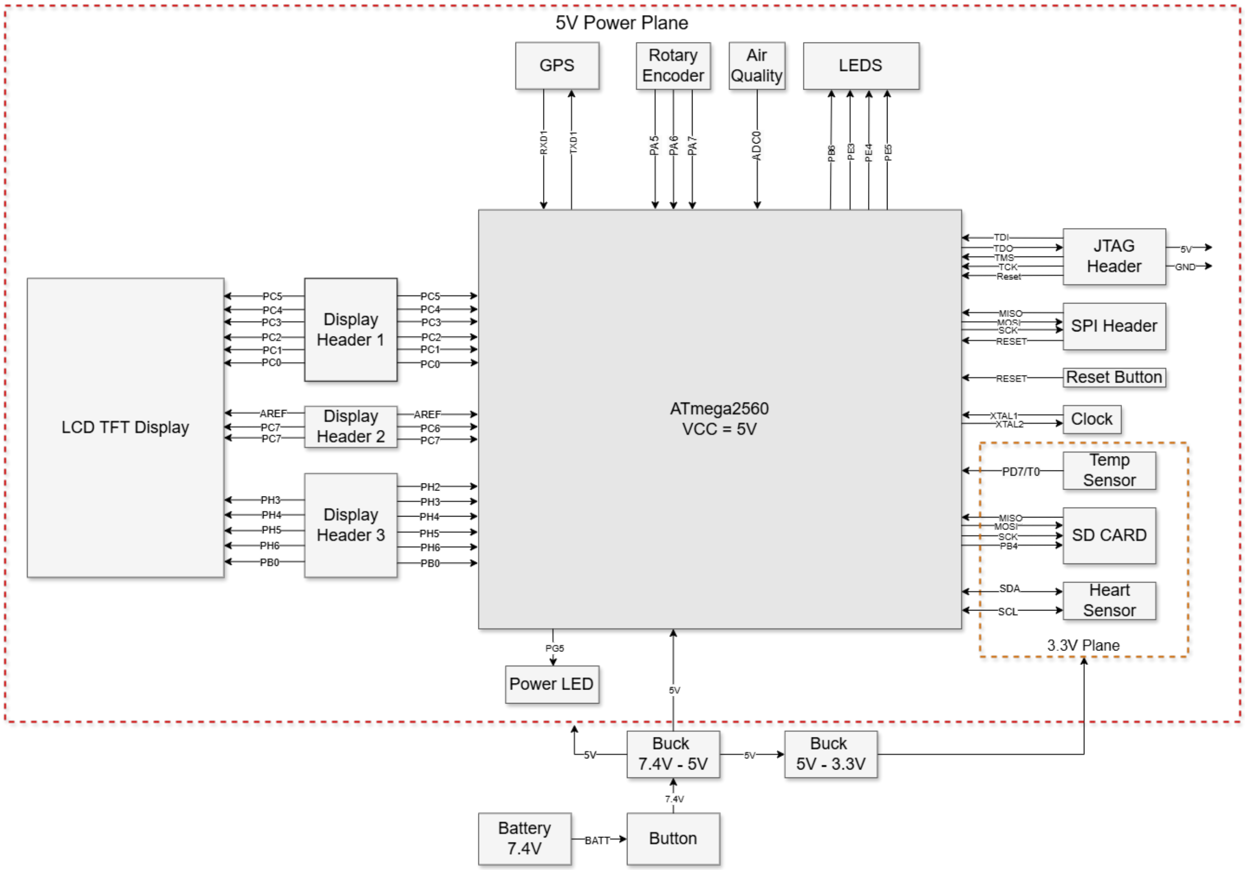
The E-P.I.P. (Environmental and Personal Information Processor) is a safety device designed for harsh environments. It uses various sensors to keep users informed about their surroundings. The device is activated by pressing the top power button, indicated by a green LED. Upon successful initialization, the LCD screen displays start-up and information screens. Users can select from Air Quality, Temperature, Humidity, Location, and Vitals, using a rotary encoder. Sensors run in parallel, and red LEDs flash to alert users to switch to relevant screens when unsafe conditions are detected. The device includes an antenna for GPS, a power port for charging, and an onboard SD card for data storage. It is powered by a battery up to the 3V3 power domain. The E-P.I.P. is ideal for use in lithium mines, firefighting, oil rigs, and farming in third-world countries, ensuring user safety and providing critical data in emergencies. In this document, we will go over the behavioral Specification, each subsystem, and its purpose, requirements, and connections.

Table of Contents

Behavioral Specification

The E-P.I.P. (Environmental and Personal Information Processor) is a safety device designed for use in harsh environmental conditions where multiple factors may be unknown. To keep the user informed about their surroundings, the E-P.I.P. utilizes various sensors. To turn on the device the top power button must be pressed, the green LED at the top of the device signals power is on. If the system initializes correctly the LCD screen will show a start-up screen and all the available information screens to choose from. The main selections will be Air Quality, Temperature and Humidity, Location, Vitals and FM radio. A rotary encoder is used to switch through the information screens to choose from the mentioned functions. All the sensors run in parallel in the background. If one of them detects unsafe conditions or wants the user to check their vitals. There will be 4 red LEDs with the function's name on the device's casing. If the LED is flashing it means that the user should switch to that information screen. The Air Quality function will send an Alert signal if the Air quality reaches a certain threshold. The Temperature and humidity LED will flash if the temperature is within unsafe conditions. The GPS Location LED will flash when the system has lost satellite connection, storing the last known coordinates. The Vitals LED will flash red when the user is asked to take a reading of their heart rate. To get a reading of the heart rate the user must press their finger against the sensor next to the display. The back includes the antenna that connects to the GPS module and the power port to charge the device. The battery of the device will power the entire device up to the 3V3 power domain. Lastly, the device will include an onboard SD card. The device will hold a record of the user uploaded from a computer and the final vitals location and environmental data the device recorded. All in all, the main purpose of the device is to be used in a harsh environment similar to that of a lithium mine, firefighters, oil rig operators, or farming in third-world countries. A device like this would keep users safe and in the worst-case scenario understand what happened to the user. The following paragraphs detail the application of the sensors and specify the components that can be used to complete the aforementioned functions.

Block Diagram



Bill of Materials

BOM					
Mfg Part	**Description / Value**	**Designator**	**Qty per board**	**Web Page**	**Datasheet**
ATMEGA2560-16AU	IC MCU 8BIT 256KB FLASH 100TQFP	U1	1	Digikey	Data
CSTNE16M0V530000R0	Resonators 16.0 MHZ 15PF	U2	1	Digikey	Data
TPS562201DDCR	Buck Switching Regulator IC Positive Adjustable 0.768V 1 Output 2A SOT-23-6 Thin, TSOT-23-6	U3, U4	2	Digikey	Data
ERJ-3EKF1004V	1M Resistor	R1	1	Digikey	Data
RMCF0805FT10K0	10K Resistor	R2, R3, R4, R9, R10, R11, R12, R18, R19	9	Digikey	Data
RC0805FR-073K3L	3.3K Resistor	R5,R6,R7,R8	4	Digikey	Data
RMCF0805JT180R	180 Resistor	R13, R14, R15, R16	4	Digikey	Data
RMCF0805FT54K9	54.9K Resistor	R17	1	Digikey	Data
RMCF0805FT33K0	33K Resistor	R20	1	Digikey	Data
CRGCQ0805F4K7	4.7K Resistor	R21	1	Digikey	Data
CL10B104KB8NNNC	100nF Capacitor	C1, C3, C4, C5, C6, C7, C8, C12, C18, C20, C21	14	Digikey	Data
C0603C220J5GAC7411	22pF Capacitor	C2	1	Digikey	Data
CC0805KRX7R9BB104	0.1uF Capacitor	C9, C15, C17	3	Digikey	Data
CL21A106KOQNNNE	10uF Capacitor	C10, C13, C14	3	Digikey	Data
CL21A226MQQNNNE	22uF Capacitor	C11, C19	2	Digikey	Data
CL10A475KP8NNNC	4.7uF Capacitor	C16	1	Digikey	Data
TFM252012ALMA3R3MTAA	3.3uH Inductor	L1, L3	2	Digikey	Data
CV201210-100K	10uH Inductor Fixed	L2	1	Digikey	Data
M20-9980346	Connector Header 6 Pin Male	J1	1	Mouser	Data
MSD-1-A	MICRO SD CARD CONNECTOR, 9 POSIT	J2	1	Mouser	Data
PPPC141LFBN-RC	14 Position Header Connector 0.100" (2.54mm) Through Hole Gold	J3	1	Digikey	Data

PPTC041LGBN-RC	4 Position Header Connector 0.100" (2.54mm) Through Hole, Right Angle Tin	J5	1	Digikey	Data
B2B-PH-K-S	Connector Header Through Hole 2 position 0.079" (2.00mm)	J6, J7	2	Digikey	Data
10129378-908002BLF	Connector Header 8 Pin Male	JP1	1	Mouser	Data
PH1RB-04-UA	Connector Header Through Hole, Right Angle 4 position 0.100" (2.54mm)	JP2	1	Digikey	Data
TSW-102-23-F-S	Connector Header Through Hole 2 position 0.100" (2.54mm)	JP3, JP4	2	Digikey	Data
SN74LVC125APWE4	Buffers & Line Drivers Quad Bus Buffer Gate With 3-State Output	IC1	1	Mouser	Data
SD1206S100S1R0	Diode 100V 1A	CR1	1	Digikey	Data
FSM2JSMAA	SWITCH TACTILE SPST-NO 0.05A 24V	SW1	1	Digikey	Data
EN12-HS22AF30	24PPR 12MM T/H 6MM SHAFT PUSHBUTTON SWITCH	R_Encoder	1	Digikey	Data
MAX30102EFD+T	SENSOR - OXIMETER/HEART RATE I2C	MAX30102	1	Digikey	Data
LP5910-1.8DRVR	IC REG LINEAR 1.8V 300MA 6WSON	LDO1.8V	1	Digikey	Data
PPPC081LFBN-RC	8 Position Header Connector 0.100" (2.54mm) Through Hole Gold	DISPLAYHEADER1, DISPLAYHEADER2	2	Digikey	Data

Item	Description	PPU	Count	Total Price	Vendor	Data Sheet
DHT22	Temperature sensor module	$10.00	1	$10.00	Amazon	Data Sheet
MQ135	Air Quality Sensor	$0.95	1	$0.95	ElectroPeak	Data Sheet
151034RS03000	Red LED	$0.17	4	$0.68	Digikey	Data Sheet
LCD	QD3502 (ILI9486)	$18.99	1	$18.99	Amazon	Data Sheet
NEO-6M-0 GPS	GNSS GPS module	$8.99	1	$8.99	Amazon	Data Sheet
328 Adafruit	Lithium Ion Polymer Battery 3.7V 2500mAh	$14.95	2	$29.90	Digikey	Data Sheet
USB LiIon/LiPoly charger	Battery Charger for 3.7V Batteries	$12.50	1	$12.50	Adafruit	Data Sheet
RP3508BBLKGRNGRNNS	Switch Pushbutton SPST 3A 125V	$6.37	1	$6.37	Digikey	Data Sheet

Subsystem 1: MCU - ATmega2560-16AU

Purpose: Due to its high-performance, low-power 8-bit AVR RISC architecture, which balances power consumption and processing speed. With 256 KB of flash memory, 8 KB of SRAM, and 86 general-purpose I/O lines, it can handle multiple sensors and peripherals simultaneously. Its robust features, including PWM, ADC, I2C, and multiple USARTs, make it ideal for managing the various functions of the E-P.I.P., ensuring reliable data processing and communication in harsh environments.

Requirements and Specifications:

- Package: 100-TQFP (Thin Quad Flat Pack)
- Operating voltage: 5V
- Temperature range: -40°C to 85°C

Booting Mode (Pg.342):

- **Programmer**: MPLAB® PICkit™ 5 In-Circuit Debugger via four JTAG-specific pins (TCK, TMS, TDI, and TDO. Control of the reset and clock pins is not required.) with a USART or SPI interface as a fallback method.
- **Hardware**: Connect the power supply, crystal oscillator, and capacitors and ensure proper connections for the SPI interface.
- **Software**: Use MPLAB X IDE to upload code to the program device

Clock Cycle (Pg.39):

- Master full duplex/receiver mode maximum of 16MHz

Memory:

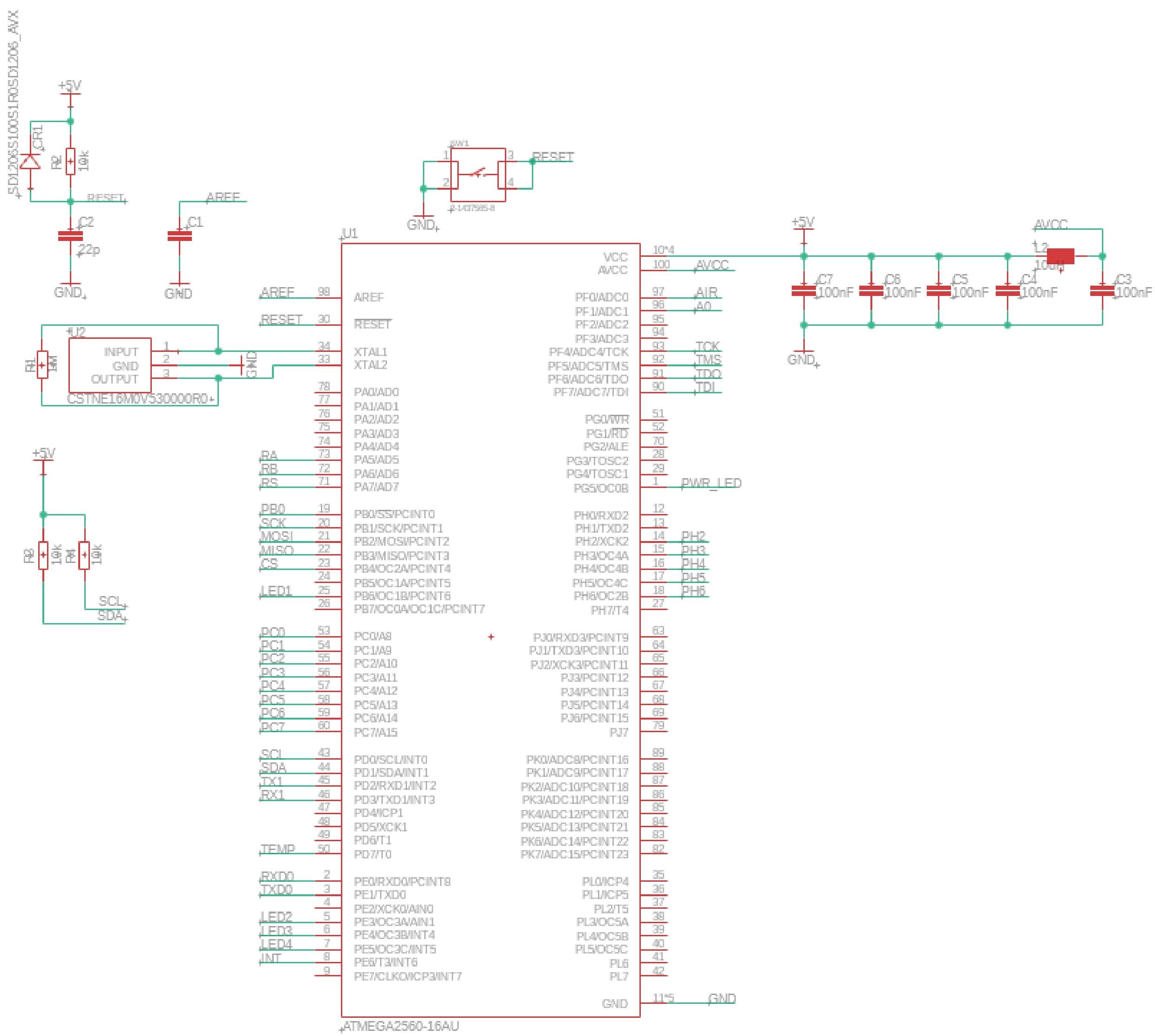
- Flash Memory: 256 KB
- SRAM: 8 KB
- EEPROM: 4 KB
- Program Memory Type: Flash
- Data Memory Size: 4 KB

System Connections:

PIN	Type	Connection
PA5	GPIO	Rotary
PA6	GPIO	Rotary
PA7	GPIO	Rotary
PF0	ADC or GPIO	AIR
PB1	SCK	MICRO
PB2	MOSI	MICRO
PB3	MISO	MICRO
PB4	GPIO, PWM	MICRO
PB6	GPIO, PWM	LED1
PE3	GPIO, PWM	LED2
PE4	GPIO, PWM	LED3
PE5	GPIO, PWM	LED4
PD0	SCL	HEART
PD1	SDA	HEART
PD2	TX1	GPS
PD3	RX1	GPS
PD7	GPIO	TEMP
PF4	ADC or GPIO, TCK	JTAG
PF5	ADC or GPIO, TMS	JTAG
PF6	ADC or GPIO, TDO	JTAG
PF7	ADC or GPIO, TDI	JTAG
PG5	GPIO,PWM	PWR_LED
PB0	GPIO	TFT
PC0	GPIO	TFT
PC1	GPIO	TFT
PC2	GPIO	TFT
PC3	GPIO	TFT
PC4	GPIO	TFT
PC5	GPIO	TFT
PC6	GPIO	TFT
PC7	GPIO	TFT

The Schematic shown below includes the bypass capacitors for the MCU, Pull up capacitors for I2C, External 16MHz Clock, and Reset Button and Pull up resistor.

MCU Schematic:



Subsystem 2: AIR QUALITY SENSOR - MQ135

Purpose: The MQ135 air quality sensor is designed to continuously monitor air quality and detect harmful gasses such as CO2, volatile organic compounds (VOCs), alcohol vapors, benzene, and smoke. This sensor helps workers assess whether their environment is safe to work in, preventing long-term health damage from poor air quality. It provides both analogs that enable real-time air monitoring and trigger alerts when gas levels exceed a safe threshold.

Requirements and Specifications:

- Operating voltage: 5V
- Current consumption: 1.0mA ($\frac{5V}{10k\Omega} = 0.5\, mA \approx 1\, mA$)
- **Communication with MCU**:
  - Analog signal: Provides real-time gas concentration levels, processed via the MCU's ADC.
- **Operating Conditions**:
  - Warm-up period: Required to reach optimal operating temperature before accurate readings can be provided.
- **Pin Configuration**:
  - Pin 1: Measurement pin
  - Pin 2: Heater voltage
  - Pin 3: Measurement pin
  - Pin 4: Measurement pin
  - Pin 5: Heater voltage and ground
  - Pin 6: Measurement pin

Interface:

- Analog Output: The MQ135 outputs an analog signal representing gas concentration, which is read via a microcontroller’s ADC (analog-to-digital converter).

Driver:

- **No special driver needed**: Outputs are processed using standard GPIO or ADC libraries.
- Standard Arduino IDE libraries for air quality monitoring

MQ135.cpp:

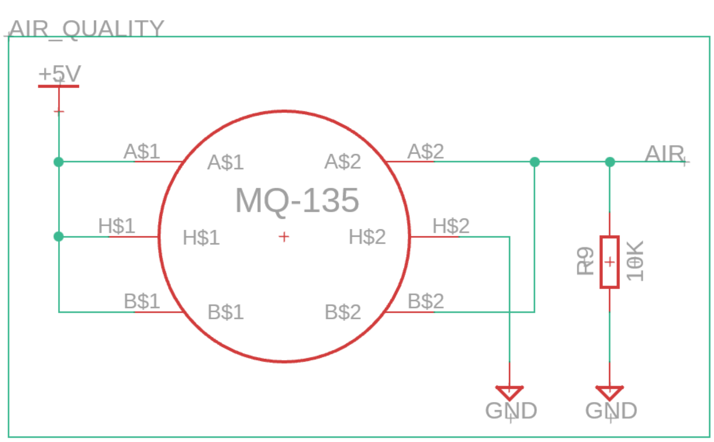
- **Sensor Setup and Calibration**:
  - The file provides methods to configure the sensor by specifying the input pin and

calibration values:

- **Pin**: Analog pin used to read sensor data.
- **rzero**: Calibration resistance at atmospheric CO2 levels.
- **rload**: Load resistance on the sensor board in kilo-ohms.

- **Correction for Environmental Factors**:
  - `getCorrectionFactor` method adjusts the sensor's resistance using different formulas for temperatures below and above 20°C.
- **Sensor Resistance Calculation**:
  - `getResistance` method computes the sensor's resistance based on analog readings from the pin.
- **CO2 Concentration Calculation**:
  - The file provides a method to calculate the CO2 concentration in ppm (parts per million) based on the corrected sensor resistance, assuming CO2 is the only gas present.

Air Quality Sensor Schematic:



The schematic for the MQ-135 air quality sensor shows it connected to a power supply, ground, and output signals. On the top left is a 5V supply that powers the sensor, while the heater circuit (HS1 and HS2) internally heats the sensor for accurate gas detection. The analog output "AIR" is fed to the ground via a 10kΩ pull-down resistor, R9, which stabilizes the signal by damping the noise. This output represents the gas concentration levels detected by the sensor and can then be read by an external microcontroller through its analog input pin. This shows the ease with which the MQ-135 sensor can be integrated into larger systems for air quality monitoring applications.

Subsystem 3: TEMPERATURE SENSOR - DHT22

Purpose: The DHT22 temperature and humidity sensor is designed to provide accurate environmental measurements, particularly useful for workers in harsh environments such as chemical plants or mines. By monitoring the humidity and temperature, the sensor can warn users if conditions reach levels that could be harmful to their health.

Requirements and Specifications:

- Power supply voltage: 3.3V
- Max current usage: 2.5mA
- **Communication with MCU**:
    - **Digital signal**: Transmits processed temperature and humidity data via a single-wire digital interface, simplifying connection to external microcontrollers.
    - **Data speed**: The sensor sends 40 bits of data (humidity, temperature, and error-checking info) at a rate of 1Hz (once per second), ensuring reliable and frequent updates.
    - Requires a pull-up resistor (typically 10kΩ) for stable communication on the data line.
- **Operating Conditions**:
    - **Temperature sensor**: NTC (Negative Temperature Coefficient) thermistor that decreases resistance as temperature rises, allowing accurate temperature calculations.
    - **Humidity sensor**: Resistive sensor that measures changes in electrical resistance due to moisture.
    - **Data transmission**: The sensor sends data at a rate of once per second (1Hz) after the microcontroller pulls the data line low.
    - **Response time**: For real-time data, it's recommended to wait 5 seconds between reads.
    - **Operating temperature**: -40°C to 80℃

- **Pin Configuration**:
  - Pin 1: VCC 3.3V
  - Pin 2: Data pin (connected to the MCU via a single wire, with a pull-up resistor)
  - Pin 3: Not connected
  - Pin 4: Ground

Interface:

- **Single-Wire Digital Interface**: Communicates via one GPIO pin for temperature and humidity data.
- **Timing-Specific Protocol**: Requires precise timing for data exchange between the sensor and microcontroller.
- **Microcontroller Compatibility**: Easily connects to platforms like Arduino or Raspberry Pi via a GPIO pin.

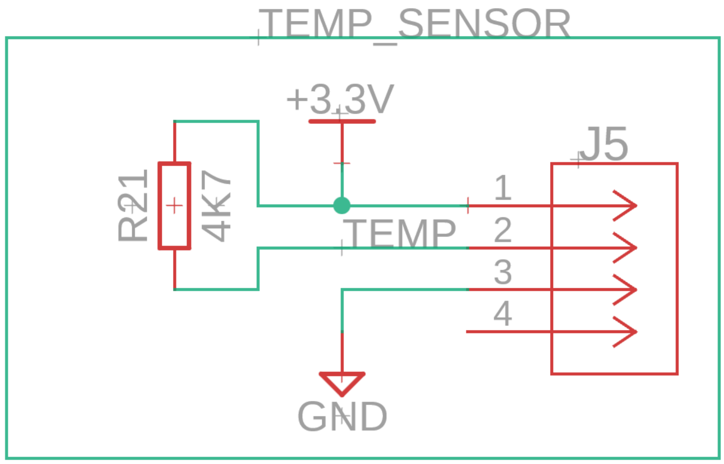
Driver - DHT.cpp:

- **DHT Sensor Implementation**:
  - This file is responsible for controlling DHT series sensors (e.g., DHT11, DHT22), which are used to measure temperature and humidity.
  - It includes sensor setup, data reading, and processing functionalities.
- **Dependency on Adafruit Sensor Library**:
  - The DHT.cpp file requires the Adafruit Unified Sensor library for compatibility with other sensor types.
  - This is explicitly mentioned in the introduction of the file, which states that you need to install the Adafruit Unified Sensor Library to use this DHT class

Adafruit_Sensor.cpp:

- **Unified Sensor Framework**:
  - The file is part of the Adafruit Unified Sensor library, which provides a standard interface for various types of sensors.
  - It defines methods and structures that represent sensor data in a consistent format across different sensors.

Temperature Sensor Schematic:



The following schematic is for a DHT22 temperature and humidity sensor. Included in this is a pin header, which is shown by the component labeled "J5." This is a 4-pin header physically used to connect this DHT22 sensor with other circuitry or a microcontroller. So, the pin layout includes supply (+3.3V), ground (GND), and a data signal (TEMP), plus one extra for flexibility. The header allows plug-and-play integration, easy assembly, and maintenance of the project that requires temperature and humidity monitoring. This module is very useful for prototyping and field applications.

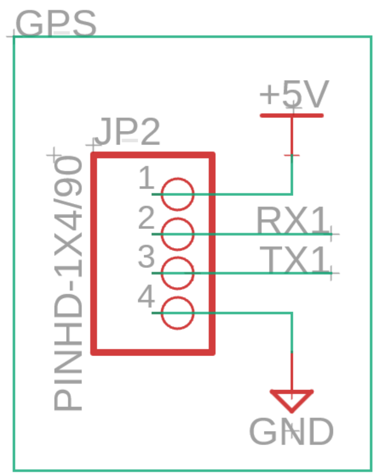
Subsystem 4: GPS - NEO-6M GPS

Purpose: GPS receivers can play a crucial role in tracking the user's location. The NEO-6M GPS module utilizes its GPS input signal external antenna to receive the module's current position based on its longitude and latitude through satellite positioning. It can be very useful for outdoor activities, like hiking or geocaching, or for navigating around harsh terrains. In emergencies, it works together with the heart rate monitor and air quality sensor. The NEO-6M GPS module will record the user's last known coordinates into the MicroSD when vital signs reach critical levels, such as exhibiting abnormal heart rate or air pollutants detected.

Requirements and Specifications:

- Operating voltage: 2.7V to 5.0V, with a 3.3V Low Dropout Regulator on board
- Operating current: 50mA
- Operating temperature: -40°C to 85°C
- **Communication with MCU**:
  - Uses **UART** interface for serial communication
  - Default baud rate of 9600 bps
  - Data transmission: set default update rate of 1 Hz
- **Pin Configuration**:
  - Pin VCC: 5V
  - Pin RX: TXD; TX pin is defined
  - Pin TX: RXD; RX pin is defined
  - Pin GND: Ground
- **Device Drivers**: The following GitHub repositories contain libraries for the GPS module that need to be imported in order to function properly with the system.
  - SerialSoftware.h
    - Establishes serial communication with the GPS module and outputs raw data in NMEA (National Marine Electronics Association) format, the standard GPS language
  - TinyGPS++.h
    - TinyGPSPlus is a *NEW* library for parsing NMEA data streams provided by GPS modules
    - Converts NMEA messages into a readable and useful format

GPS Schematic:



The schematic shown above is a right angle 1x4 male pin header that connects to the NEO-6M GPS module via through hole technology. Pin 1 connects to 5V VDD pin, Pin 2 connects to the RX or receiver pin and Pin 3 connects to TX or transmitter pin. Pin 4 connects to GND pin.

Subsystem 5: VITALS - MAX30102

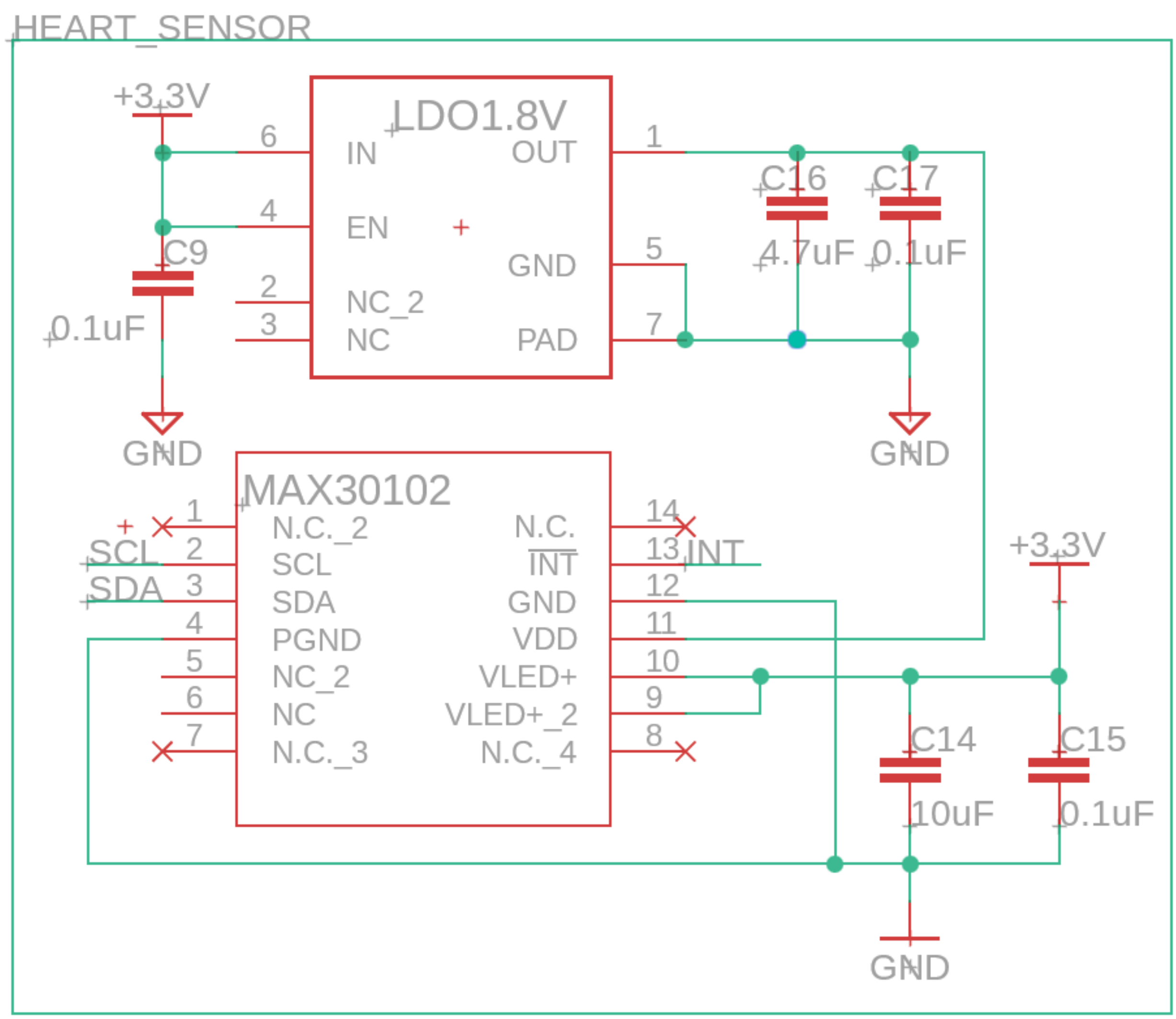
Purpose: The MAX30102 heart rate sensor is one of many sensors within the E-P.I.P handheld device that works by monitoring the user's vitals specifically with their heart rate as well as their blood oxygen levels which would help diagnose the user's vitals if they are located in potentially unsafe environments. A couple of examples of the heart rate sensor/ blood oxygen sensor being used would be a situation where one is working with coal mines or other oxygen-limited environments, the device would prompt the user to check their vitals using this sensor on the E-P.I.P which would then provide the user with heart rate information as well as blood oxygen saturation. The device would then alert the user of potential abnormal vital conditions such as an abnormal heart rate (too high or too low), or even dangerously low blood oxygen levels (less than 90 percent).

Requirements and Specifications:

- Operating voltage: 3.3V for LEDs, 1.8V for I2C interface
- Current: 600mA
- Decoupling power to prevent noise from other sources
- **Communication with MCU**:
    - I2C interface: standard 12C speeds at 100kHz with speeds up to 400kHz
- **Operating Conditions**:
    - Operating Temp: -40 to 85 degrees Celsius
    - Humidity Range: 10% to 95% RH
- **Required External Components for Sensor (on PCB)**:
    - Pull up resistors: for I2C connections around 4.7 - 10k ohms
    - Decoupling Capacitors: capacitors to reduce noise and cleaner power
- **Pinout**:
    - Pin VDD: 1.8V
    - Pin VLED+: 3.3V
    - Pin GND: Analog ground
    - Pin PGND: LED ground
    - Pin SCL: I2C Clock input
    - Pin SDA: I2C Clock Data
    - Pin INT Active low interrupt (Open Drain)

- **Driver** [**Sparkfun MAX30102 Library**]:
  - Responsible for controlling the sensor to measure heart rate and SPO2
  - Includes setting up the sensor, reading the data/ processing the data

Heart Sensor Schematic:



Subsystem 6: LCD - QD3501 (ILI9486)

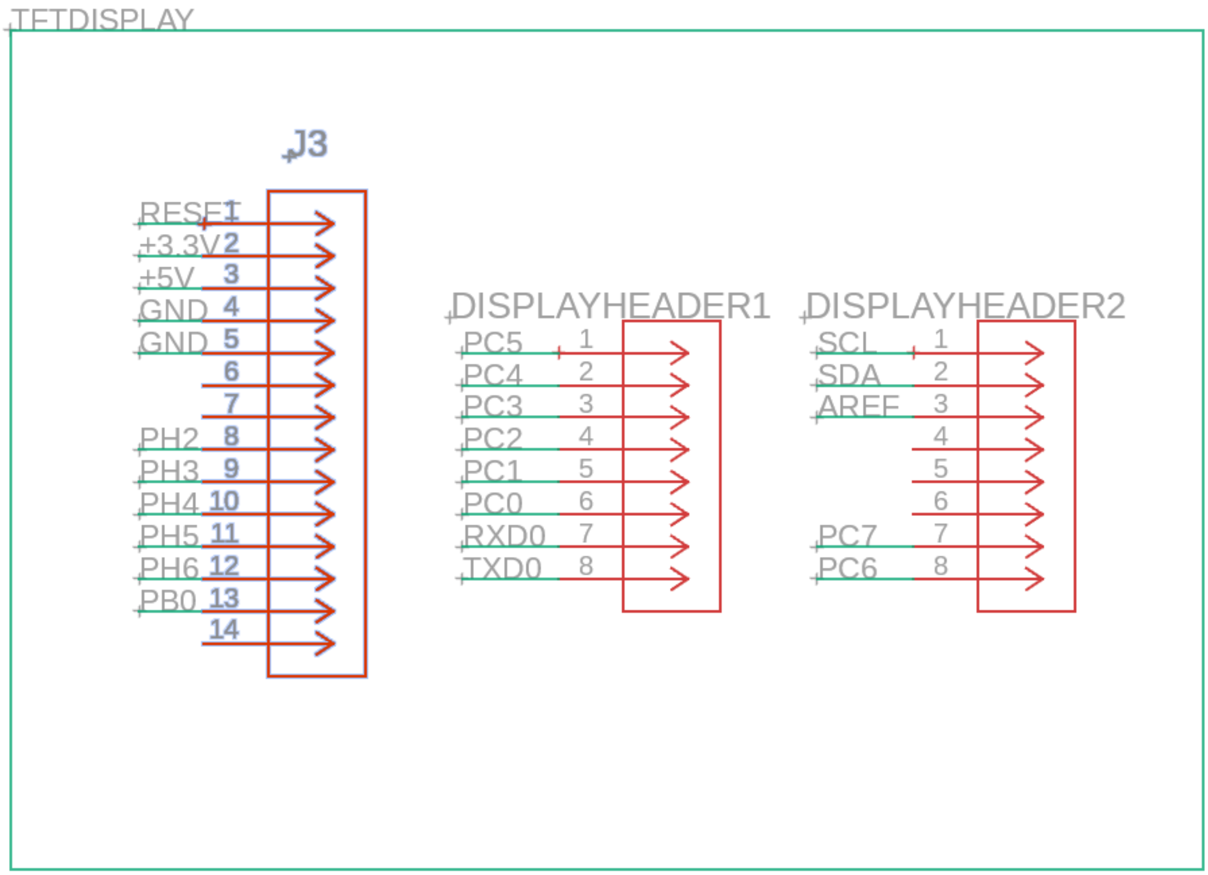
Purpose: The QD3501 TFT LCD module will be the main method for displaying our sensors and other miscellaneous information (such as time, date, and graphics). The module has a 3.5-inch display and a resolution of 320x480 pixels, the screen is large enough to visualize the sensor information and allow navigation through the multiple menus within our software.

Requirements and Specifications:

- Operating Voltage: 2.8V for typical usage, 2.5-3.3V range
- **Operating Conditions**:
  - Operating Temp: -20 - 70 degrees Celsius
  - Storage Temp: -30 - 80 degrees Celsius
  - UV Protection: protection against direct sunlight
- **Communications with MCU**:
  - Parallel connection with MCU
- **Pinout**:
  - Pin LEDA: Anode power input for backlight
  - Pin LEDK: Cathode power input for backlight
  - Pin GND: Ground pin
  - Pin VCC: 3.3V
  - Pin IOVCC : 5V
  - Pin CS: Chip select input
  - Pin WR: Write strobe signal input
  - Pin RS: Read strobe signal input
  - Pin RD: Data select input
  - Pin RST: Reset input
  - Pin DB15-DB0: Data Bus
  - Pin FMARK: Tearing Effect output signal
  - Pin XL: X-
  - Pin YD: Y-
  - Pin XR: X+
  - Pin YU: Y+
- **Driver Adafruit_TFTLCD.cpp**:

- ○ Responsible for controlling display output
- ○ Initializes pins, display resolution, and other required initialization

LCD Schematic:



Subsystem 7: DATA - Micro SD Card

Purpose: The Micro SD card provides reliable data storage and retrieval capabilities. This allows the device to store critical environmental and personal data, ensuring that information is preserved even in harsh conditions. It supports both Micro SD and Micro SDHC cards, making it versatile for various storage needs.

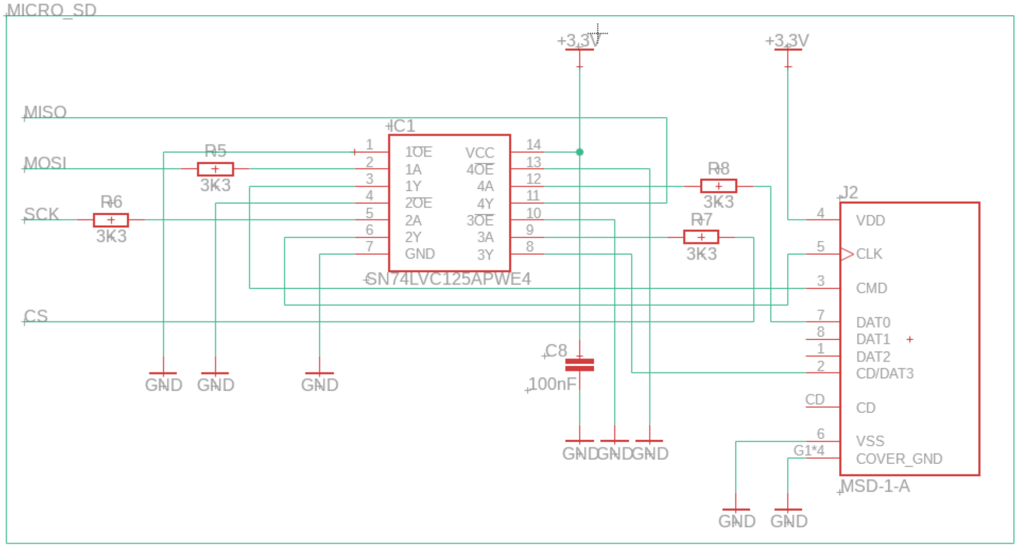
Requirements and Specifications:

- Operating voltage: 3.3V Domain
- Communication interface: Standard SPI interface
- **Pin Configuration**:
  - SPI:
    - MISO (Master In Slave Out)
    - MOSI (Master Out Slave In)
    - SCK (Serial Clock)
  - GPIO: Chip Select (Digital Pin)

Device Drivers: The following repositories hold the libraries for the device drivers for a Micro SD card to function with the system.

- SD by Arduino, Sparkfun
  - Once an SD memory card is connected to the SPI interface of the Arduino board you can create files and read/write on them. You can also move through directories on the SD card
- SDfat Arduino FAT16/FAT32 exFAT Library
  - The underlying library used to actually initialize and operate the system

Schematic:



Approved warning on schematic for J2 pin connections for power and ground, asking for signal names we did not utilize.

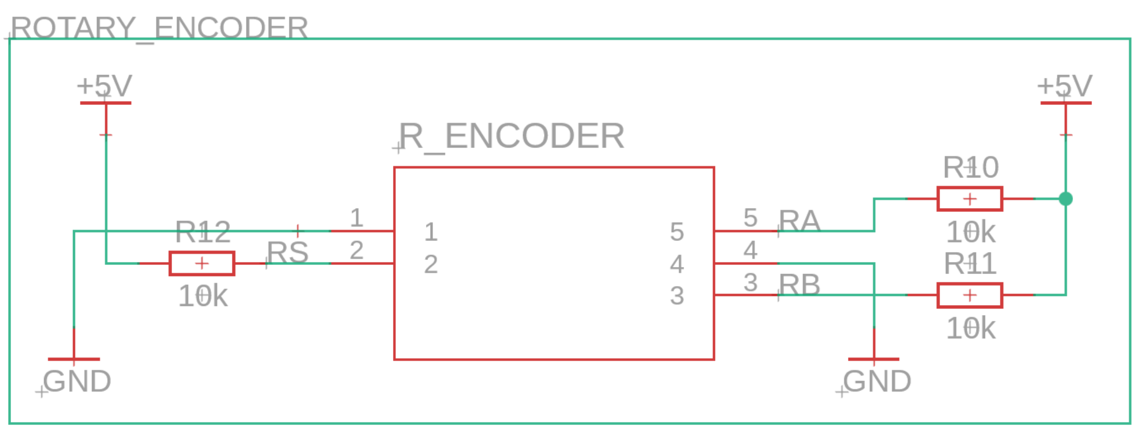
Subsystem 8: I/O
Rotary Encoder

Purpose: The rotary encoder allows the user to navigate through the various information screens of the E-P.I.P. device. It converts the angular position or motion of the knob into digital signals, enabling precise selection and control of the device's functions, even in harsh conditions.

Requirements and Specifications:

- Power Supply: 5V
- Communication Interface: GPIO Digital signals
- **Pin Configuration**:
  - GND: Connect to the ground pin of the Arduino
  - VCC: Connect to the 5V pin of the Arduino
  - CLK (Clock): Connect to a digital pin on the Arduino
  - DT (Data): Connect to a digital pin on the Arduino
  - SW (Switch): Connect to a digital pin on the Arduino

Schematic:



The following schematic represents the rotary encoder, including all the connections the device requires to operate and interface with a circuit. Here, the rotary encoder is labeled as "R_ENCODER" and is powered with 5V connected to Vin and GND. It includes three main pins: RS (switch), RA, and RB-rotary output signals that convey position data regarding the rotation and pressing of the encoder button. Pull-up resistors-R10, R11 and R12, all 10kΩ-serve to stabilize the signals by pulling up, thus avoiding any floating state and ensuring output cleanliness and stability. The RS pin supports the encoder for the push-button functionality, RA and RB provide digital signals about rotation direction and absolute position. This schematic application underlines the encoder functioning as a user interface element related to navigation or control, normally done in embedded systems.

LEDs

Purpose: LEDs provide visual alerts to the user, indicating the status of various functions such as air quality, temperature, GPS connection, and vitals. They ensure that critical information is easily noticeable, enhancing the safety and usability of the device.

Requirements and Specifications:

- Power Supply: 2V to 3.6V
- Communication Interface: GPIO
- **Pin Configuration**:
  - Anode: Connect to a current-limiting resistor and then to a digital pin
  - Cathode: Connect to ground

<u>Subsystem 9: POWER SUPPLY - 7.4V</u>

Purpose: The main power supply for all the subsystems.

Subsystem	Name	Operating Voltage (V)	Operating Current (mA)	Power (W)
MCU	ATmega2560-16 AU	5.0	200	1.000
GPS	NEO-6M GPS Module	5.0	50	0.250
LCD	QD3501	5.0	120	0.600
AIR QUALITY	MQ135	5.0	1	0.005
I/O	Rotary Encoder	5.0	1	0.005
	Red LED x 4	5.0	100	0.500
POWER	BUTTON, LED	5.0	20	0.100
VITALS	MAX30102, 3.3V to 1.8V	3.3	1.2	0.004
DATA	Micro SD Card Module, BUFFER	3.3	600	1.980
TEMP	DHT22	3.3	2.5	0.008

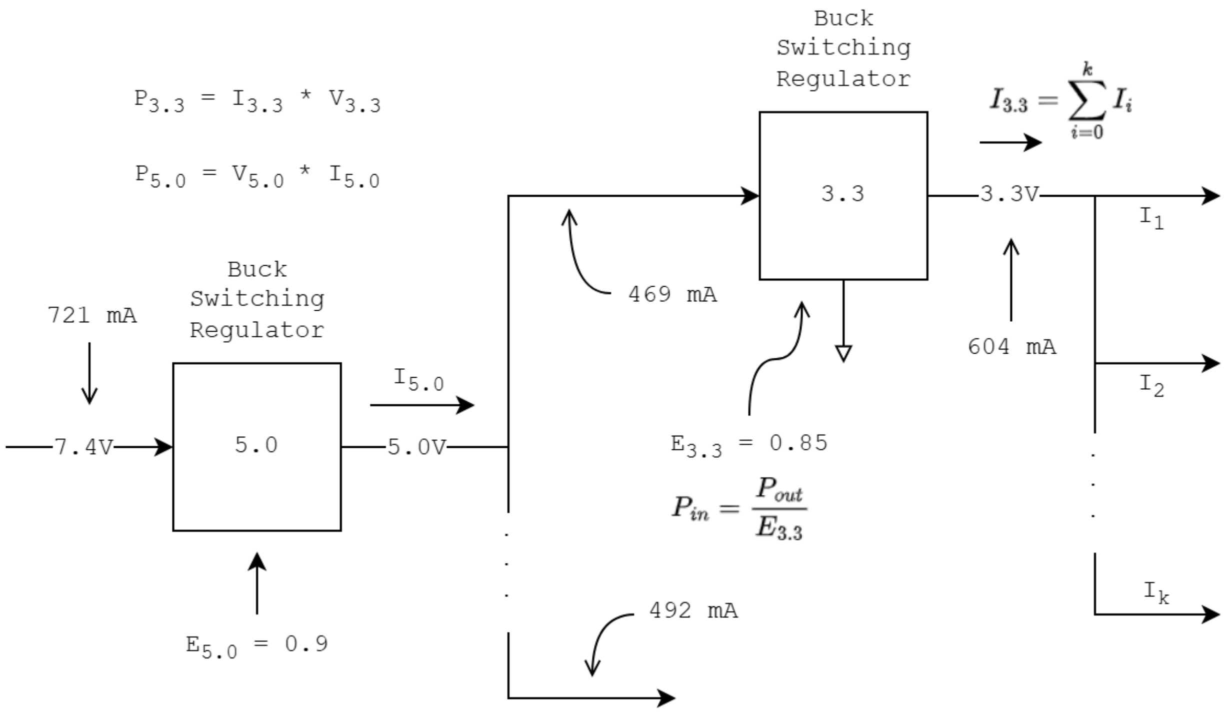
Specifications:

- Domains: 3.3V & 5.0V
- Estimated Total Current: 1.096 A
- Estimated Total Power: 4.452 W

Main Components:

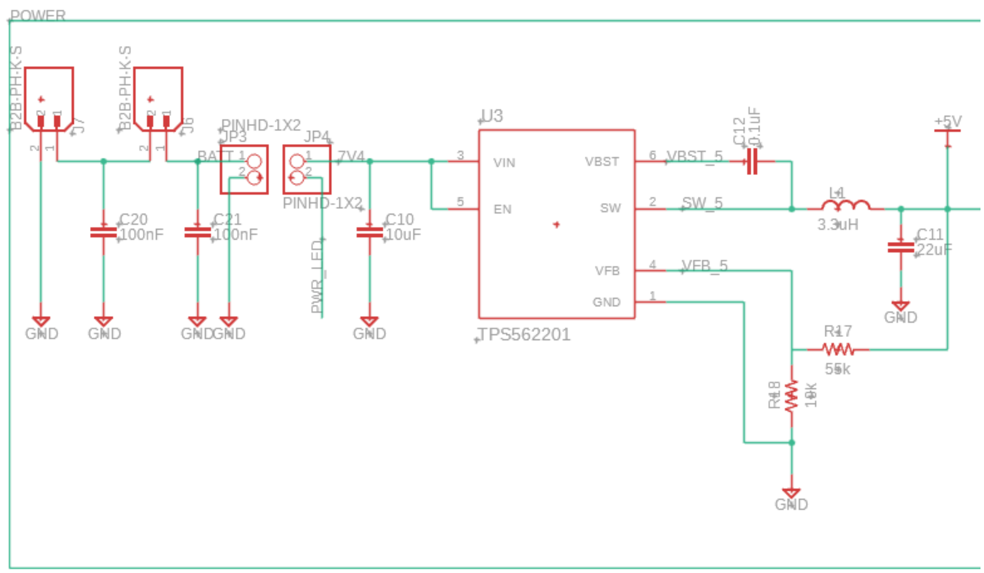
- [328 Adafruit](): Lithium Ion Polymer Battery 3.7V 2500mAh
  - Current capacity: 2500mAh
  - 2 batteries in series
    - $3.7V * 2 = 7.4V$
- [TPS562201DDCR](): Buck Switching Regulator, 2A Output (x2)
  - 7.4V → 5.0V; 90% Efficiency
  - 5.0V → 3.3V; 85% Efficiency

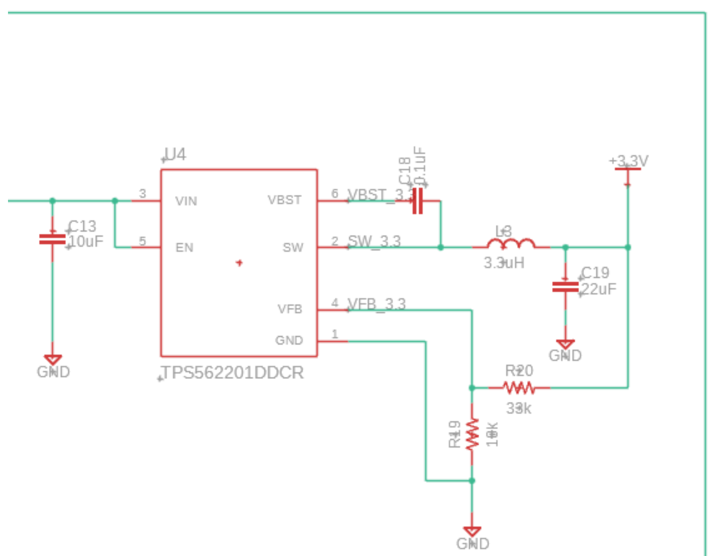
Measuring Current:



Variable	Value	Unit
3.3V Current Out:	0.6037	A
3.3V Power Out:	1.9922	W
Power In :	2.3438	W
Load Current :	0.4690	A
5V Current Out:	0.9610	A
5V Power Out:	4.8050	W
Power In :	5.3389	W
Load Current :	0.7210	A

Power Schematic:

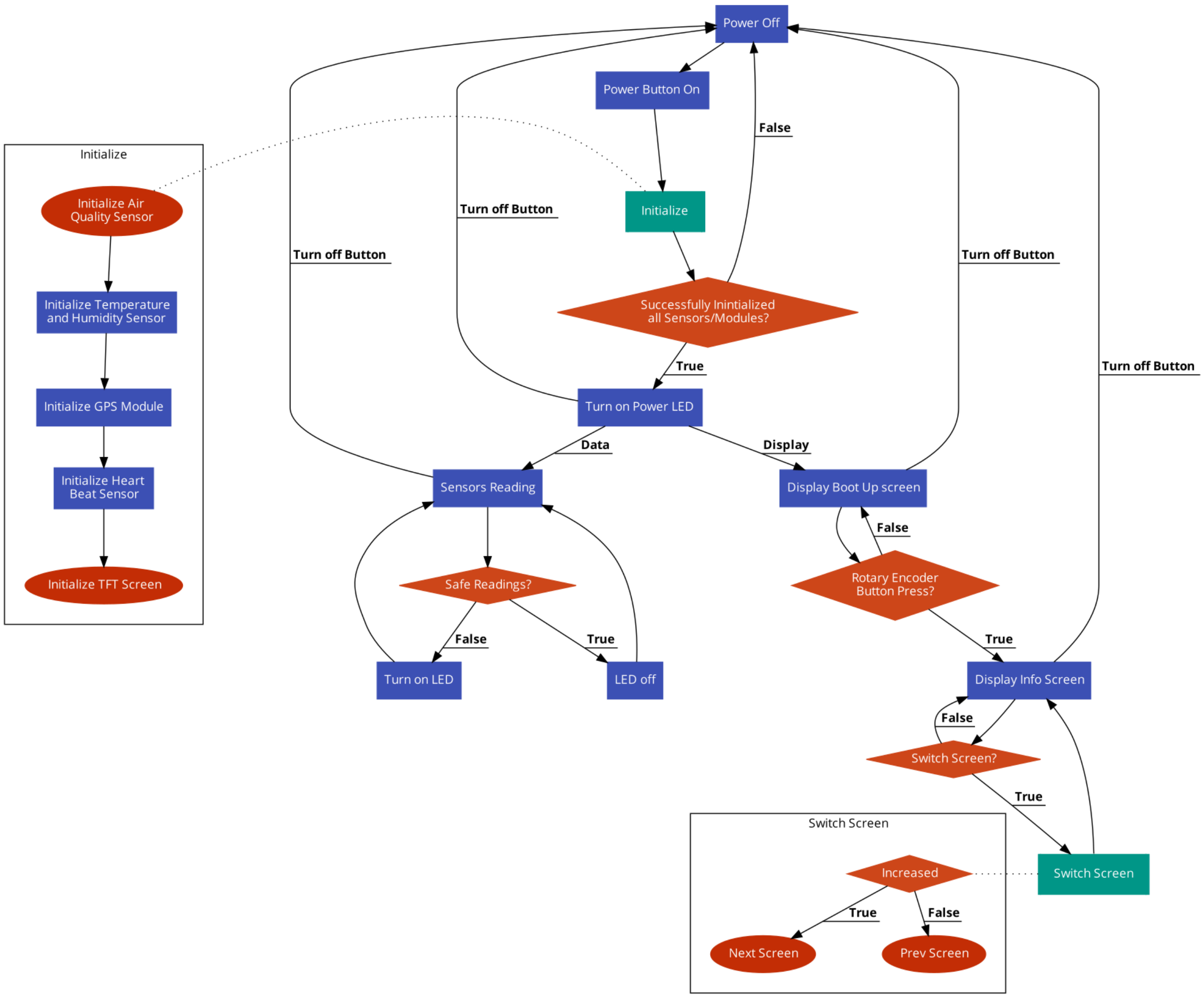




Approved warning pop-ups like "POWER pin U4 connected to +5V" because Fusion 360 expects the power pin to connect to a supply pin of the SAME name. Thus, we can ignore these.

Subsystem 10: SOFTWARE

Code Workflow

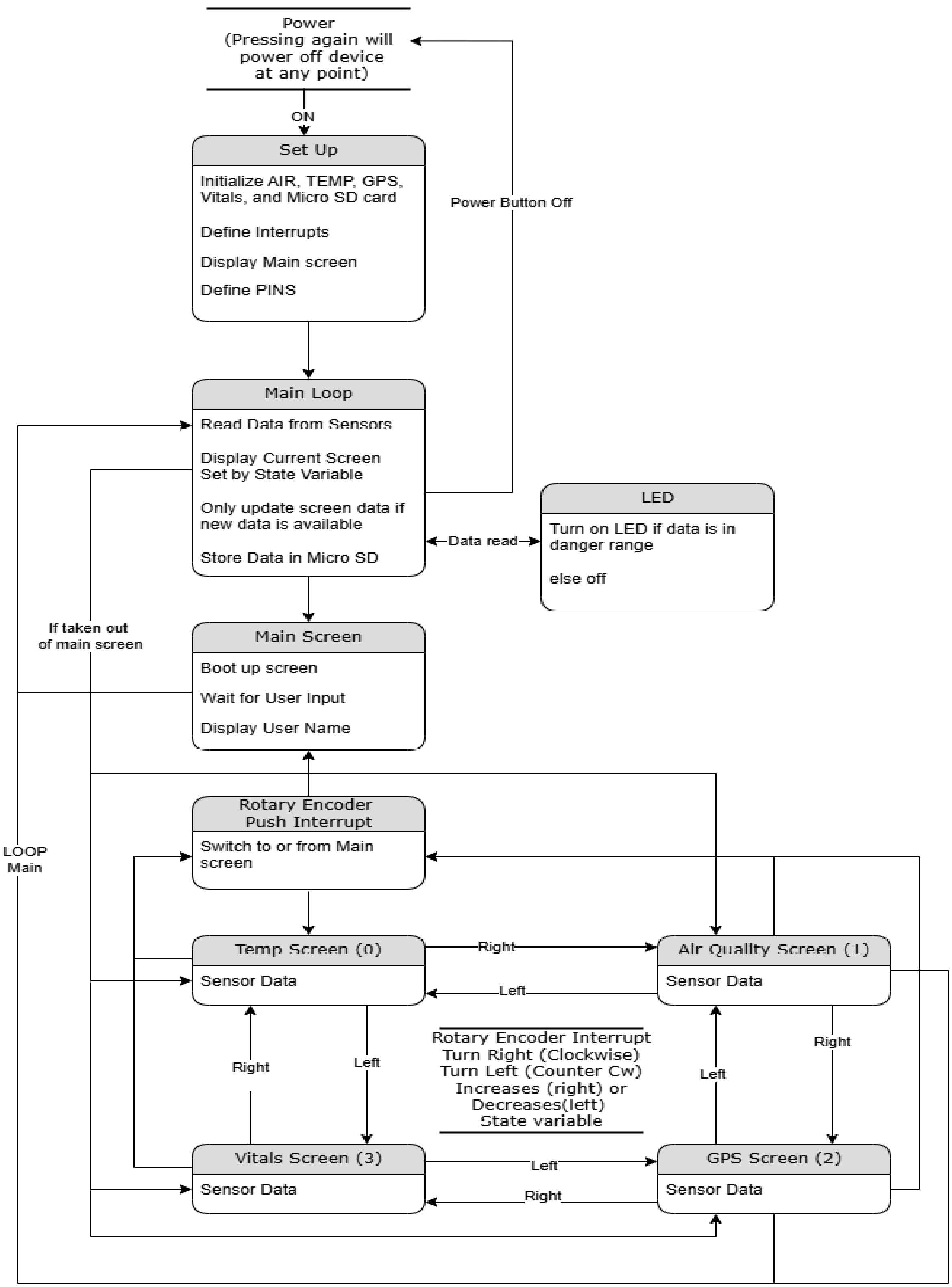


Pseudo Code

```
function Initialize(){
 Initialize Air Quality Sensor;
 Initialize Temperature and Humidity Sensor;
 Initialize GPS Module;
 Initialize Heart Beat Sensor;
 Initialize TFT Screen;
}

function Switch Screen(){
 if(Increased){Next Screen;}
 else{Prev Screen;}
 return;
}

Press Power Button;
call Initialize;
if(Successfully Initialized all Sensors/Modules?){
 Turn on the Green LED;



 branch(cross_a) [Display] {
 before_button:
 Display Boot Up screen;
 if(Rotary Encoder
  Button Press?){
  before_check:
  Display Info Screen;
  if (Switch Screen?){
   call Switch Screen;
   loop before_check;
  }
  else{loop before_check;}
  }
 loop before_button;
 }
```

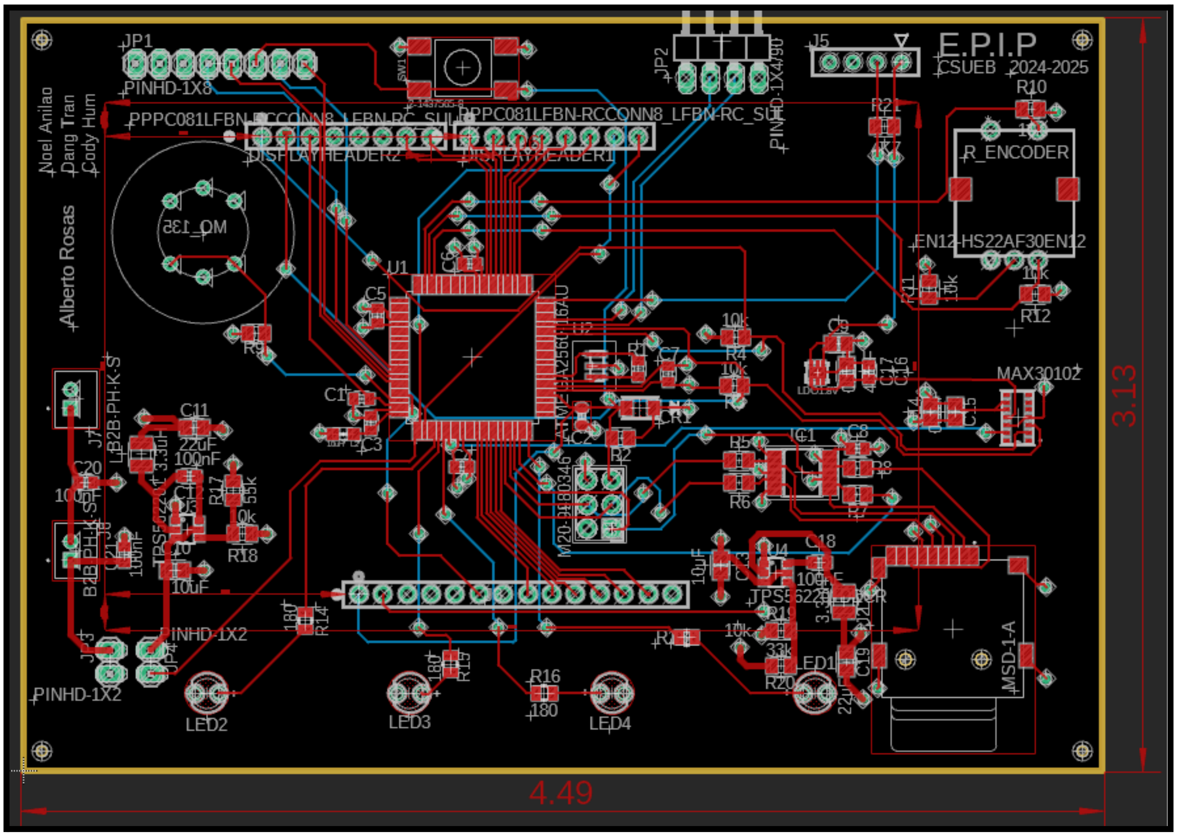
```
  branch(cross_b) [Data] {
  before_read:
  Sensors Reading;
  if(!Safe Readings?){Turn on Red LED;}
  loop before_read;
  }
  branch(cross_c) [Power Button] {Power off}
}

else(Critical Failure Power Off);
return;
```

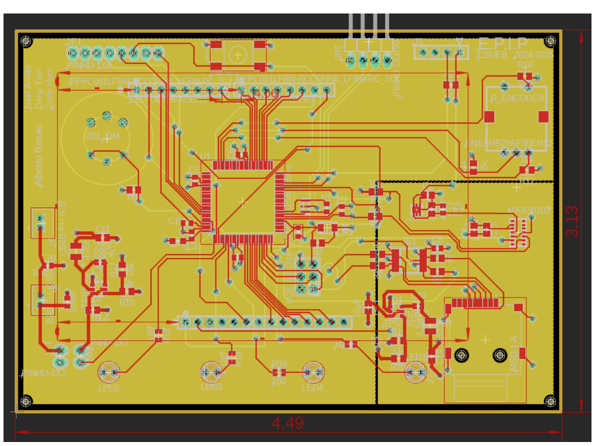
Software Diagram

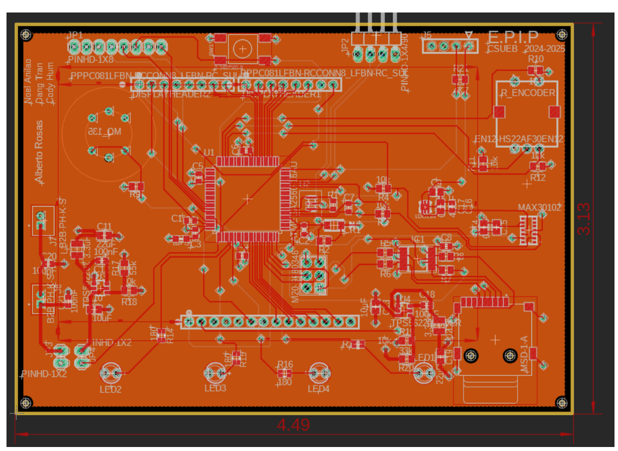


Layout

The layout design for the E-P.I.P. project is carefully crafted, with all components placed on one side of the PCB for simplicity and efficient use of space, except for the MQ135 sensor, which is positioned separately. The design consists of a 4-layer PCB, including a top layer, power plane, ground plane, and bottom layer. The top layer is dedicated to signal routing, while the bottom layer primarily handles the blue signal wires. The power plane is divided into two voltage domains: a 5V domain covering most of the area and a 3.3V domain that powers components in the bottom right corner. The routing and wiring presented a challenge, but overcoming it provided valuable experience in PCB design, ensuring that the final board is free of errors and optimized for performance.







Conclusion

The E-P.I.P. project, initially daunting, has evolved into an engaging and rewarding endeavor. The ATmega2560 MCU provides the necessary computational power and storage, while the sensors (MQ135, DHT22, NEO6M, MAX30102) continuously monitor environmental and personal data. The rotary encoder and LEDs facilitate user interaction and alerting. The GPS module ensures accurate location tracking, and the SD card module offers reliable data storage. The battery powers the entire system, ensuring functionality in harsh conditions. Detailed schematics and software development are underway, with prototype components being tested. We eagerly anticipate the E-P.I.P. device becoming fully operational, enhancing safety and providing critical data in challenging environments.