

0.1 Hardware

0.1.1 Diagrama de bloques (Hardware)

A continuación se muestra el diagrama en bloques del Hardware y los delimitadores de las secciones, siendo estas:

- Potencia
- Cargador
- Sensado

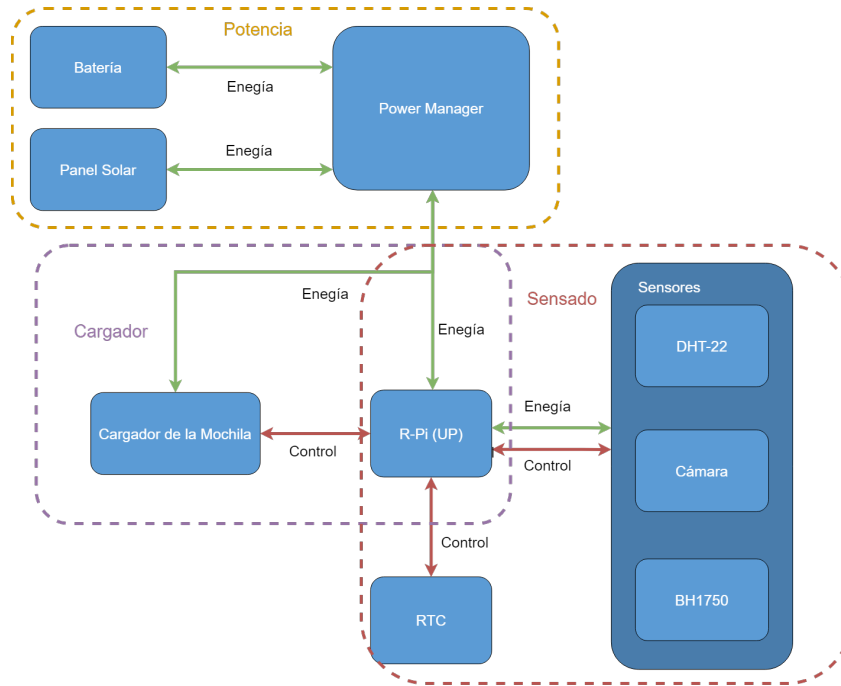


Figura 0.1.1.1: Diagrama en bloques del sistema de hardware.

La etapa de potencia es el conjunto de elementos necesarios para proveer de energía a toda la electrónica del proyecto.

El cargador es la etapa que se encarga, como su nombre indica realizar la carga inalámbrica de la UBM.

Y la de sensado se encarga de la medición de las variables físicas y de su correcto almacenamiento, teniendo en cuenta que esto implica un conocimiento preciso de la hora.

0.1.2 Descripción detallada de cada bloque

0.1.2.1 Potencia

Para la unidad de potencia se utilizarán paneles solares y una batería de gel de ciclo profundo conectados a la placa DFR0580 la cual se encarga de cargar la batería con los paneles solares y asegurar 4 salidas de tensión:

- 5V 2.5A [USB] x 2
- 5V 5A x 1
- 12V 8A x 1

, Con estas salidas se alimentarán todos los módulos, excepto por el oscilador de potencia, para este se necesita una etapa DC-DC para elevar la tensión. Esta etapa es una fuente switching de topología Boost.

0.1.2.2 Cargador

Para la etapa del cargador, la transmisión inalámbrica va a constar: del lado del transmisor, de un oscilador en 915Mhz que está comandado por la RPi, y un amplificador de potencia alimentado por un DC/DC de 12V a 15V.

Por el lado del receptor, se encuentra el integrado P1110B que estabiliza la energía para proveer realizar la carga de al UBM.

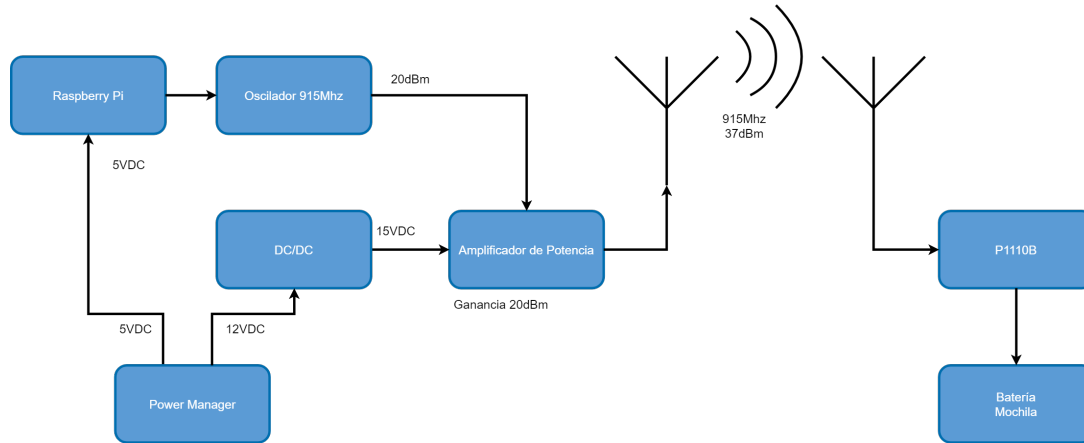


Figura 0.1.2.2.1: Diagrama en bloques cargador.

0.1.2.3 Sensado

En esta sección aparte del conexionado de los sensores haremos detalle en los pines de la raspberry pi que serán utilizados.

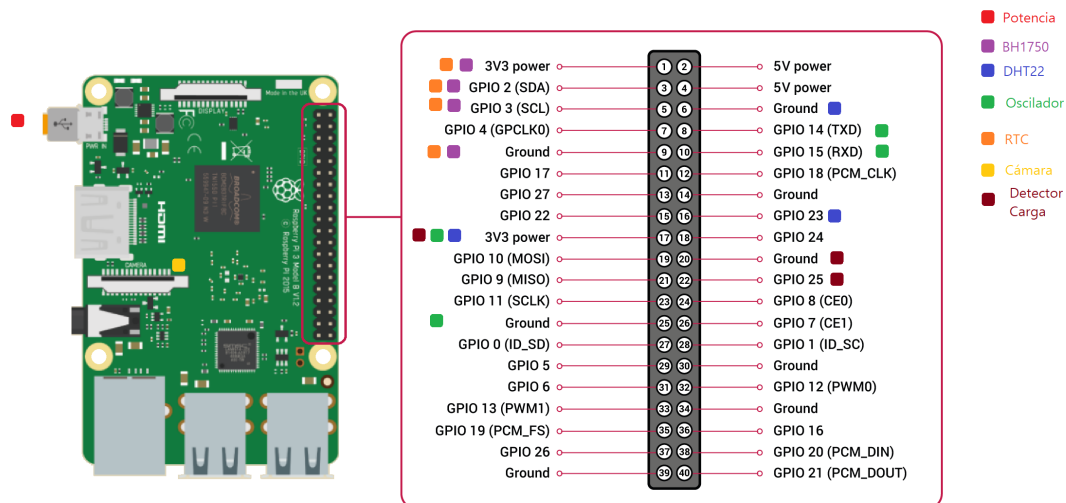


Figura 0.1.2.3.1: Conexionado Raspberry Pi.

El sensor de humedad y temperatura DHT-22 se comunica de manera serial a través de un pin de GPIO. Para que esto sea posible el sensor necesita 3.3V, GND y el pin de GPIO por donde se realiza la comunicación, teniendo en cuenta que es necesario un pull-up entre la línea de datos y 3.3V.

El oscilador necesita alimentación, tierra y su comunicación utilizará el protocolo UART.

La cámara simplemente se conecta al zócalo destinado para este propósito en la placa.

Además se tiene un pin de GPIO el cual indicará el momento de prendido del cargador.

Para el conexionado con el sensor de luminosidad y RTC se utilizará el protocolo de comunicación I^2C en modo multi-slave

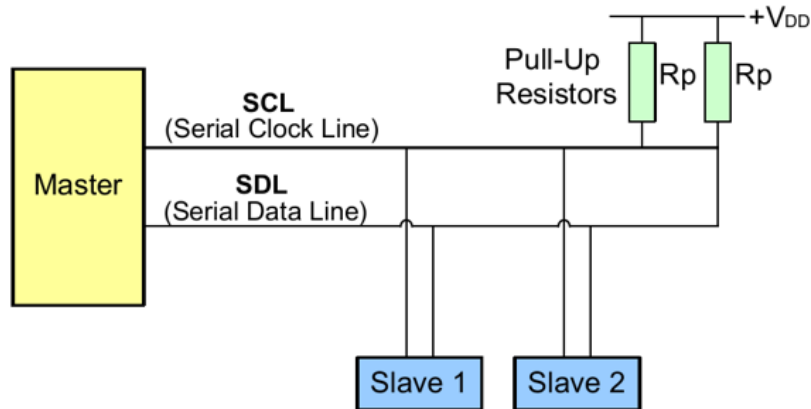


Figura 0.1.2.3.2: Conexión I^2C Multi-slave.

0.1.3 Detalles de selección y cálculo de los elementos circuitales de cada bloque

0.1.3.1 Cálculo potencias

Para comenzar los cálculos se asumen las siguientes variables sacadas de estadísticas propias de la zona donde se realiza el estudio.

1. Duración noche = 8:30 Horas
2. Duración día = 15:30 Horas
3. Horas de sol óptimo = 8 Horas

esto seguro conviene estructurarlo distinto pero lo dejo así por ahora

- Raspberry Pi: $P_{min} = 2W$ $P_{est} = 2.7W$ Si se multiplica por un día se obtiene la energía necesaria por día $E_{min} = 172.8KJ$ $E_{est} = 233.28KJ$

Los valores de potencia fueron sacados de

referencia a de donde sacamos el dato me imagino en el final de la tesis

0.1.3.2 Cálculo resistencias

Tanto para el Bus de I^2C como para el DHT22 se calcularon las resistencias de pull-up de la siguiente manera:

$$R_p = \frac{V_{dd}}{I_{Rp}} = \frac{3.3 V}{1 mA} = 3.3K\Omega \quad (1)$$

Acá van los cálculos de paneles, batería, de la boost si es que la hacemos o si usamos un integrado

0.1.4 Plan de pruebas de cada modulo

0.2 Software

0.2.1 Esquema general del proyecto

mi idea para esta parte era explicar que tecnologias
ibamos a usar y eso

0.2.2 Diagrama de estados y flujogramas

En este pasaje se mostrarán los diagramas de estados del sistema. Se puede notar que el sistema cuenta con 4 estados distintivos, siendo estos

- Initial
- Idle
- Charging
- Communicating

Cabe la pena mencionar que en cualquiera de los estados salvando por el inicial, el sistema estará realizando mediciones del ambiente, al igual de tener la posibilidad de comunicarse por Bluetooth con la electrónica de la mochila.¹

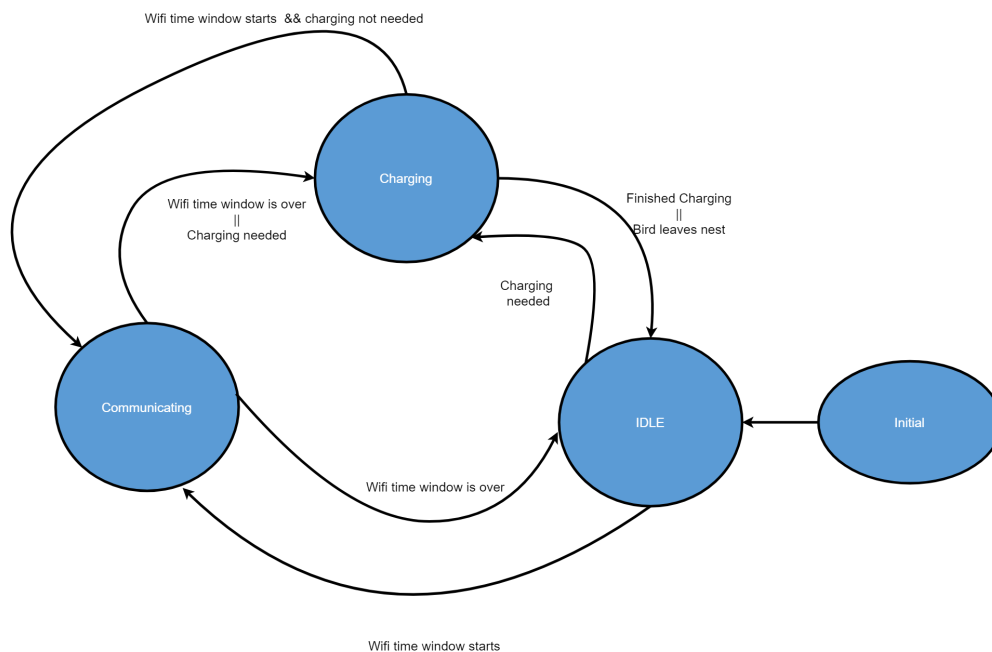


Figura 0.2.2.1: Diagrama de estados.

En el estado “Initial” es aquel donde se inicializan todos los drivers, estructuras de datos y configuraciones iniciales, no se volverá a este estado una vez que este haya sido abandonado.

¹A menos que el nivel de carga de la UBM no sea suficiente.

Initialize
State

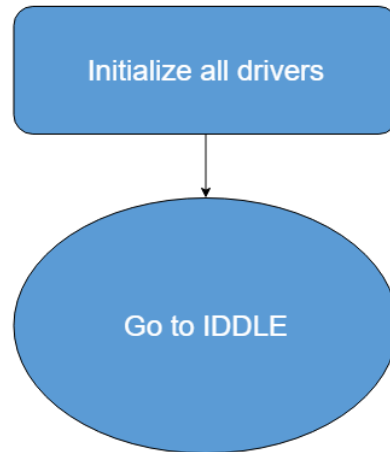


Figura 0.2.2.2: Diagrama de flujo: Estado “Initial”.

Luego se encuentra el estado “IDLE” en este estado se sensaran las variables fisicas con el periodo de muestreo acorde a cada especificado en

**PONER LUGAR DONDE ESTE ESPECIFICADO
ESTO**

luego se fijará si es momento de prender el hotspot wifi, si es asi irá al estado “Communicating”. Si no es así se verá si hay que cargar la batería, en caso positivo irá a estado “Charging”, caso contrario se corrobora si hay transmisión Bluetooth, actua acorde y comienza el ciclo nuevamente.

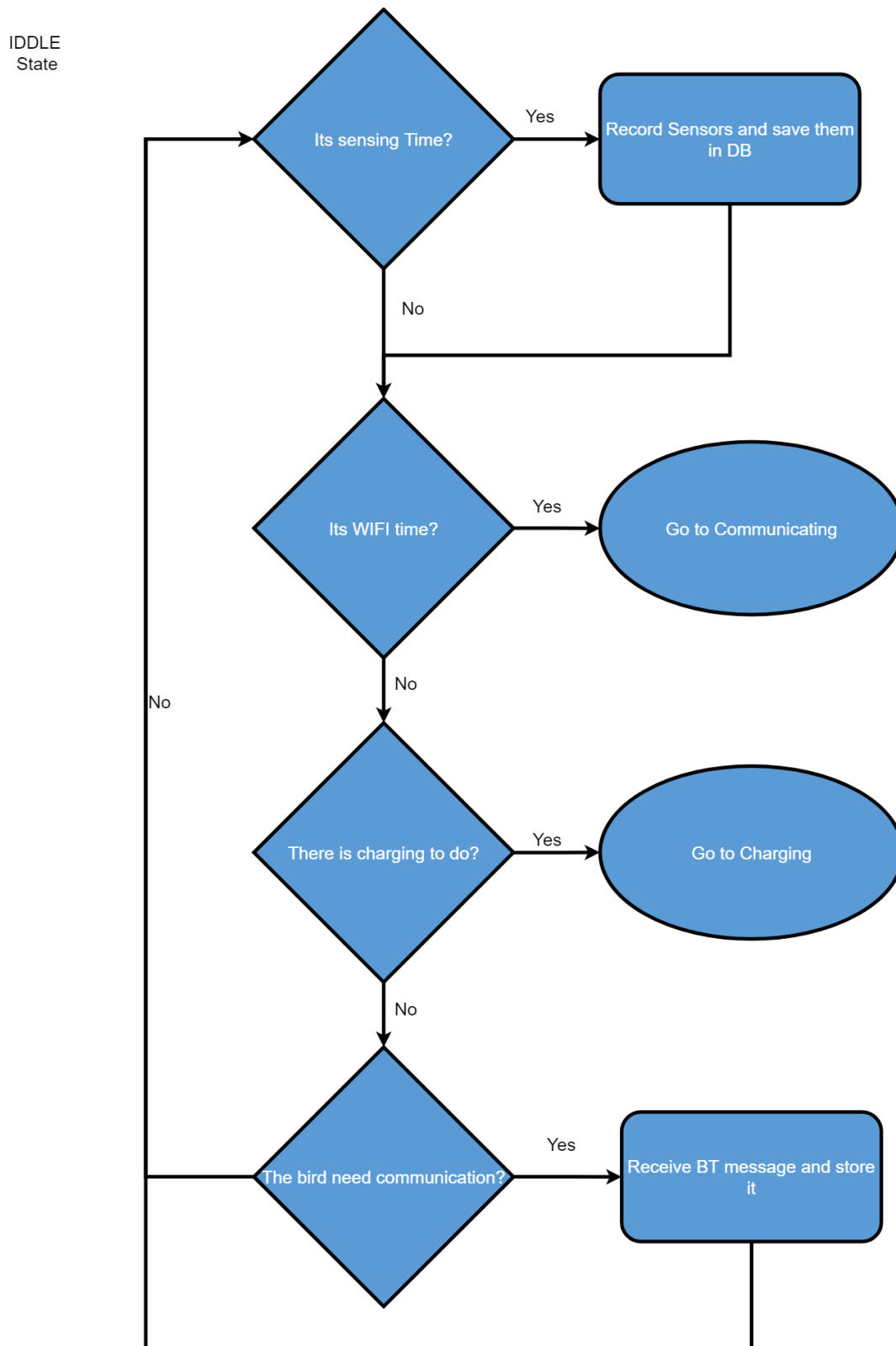


Figura 0.2.2.3: Diagrama de flujo: Estado “IDLE”.

En el estado “Charging” su función principal será la de cargar la UBM. Lo primero a hacer en este estado es habilitar el cargador, luego si hay un mensaje Bluetooth, se comienza la comunicación y se recibe hasta que esta haya terminado. A continuación se encarga del sensado, luego se verifica si la carga terminó, si es así se desactiva el cargador, y dependiendo si es momento de habilitar el hotspot o no se irá al estado “Communicating” ó “IDLE”

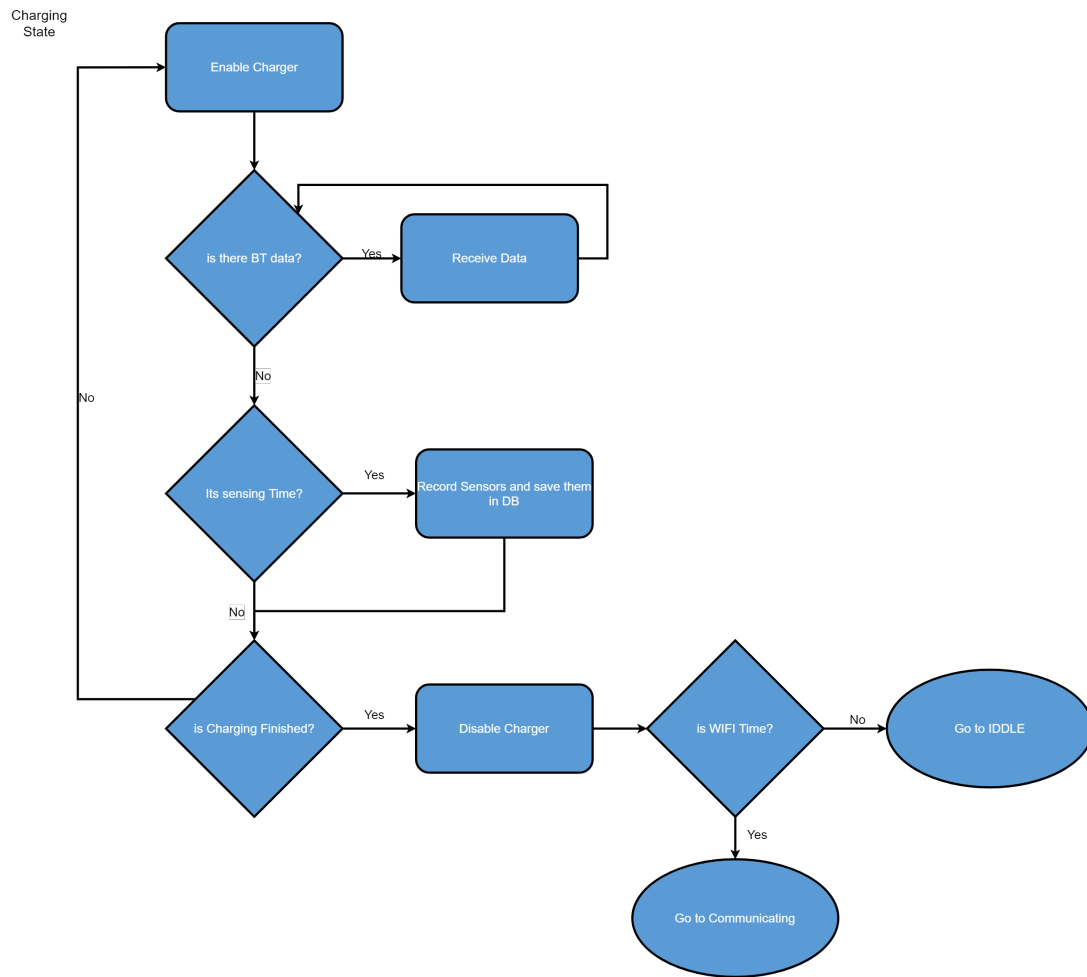


Figura 0.2.2.4: Diagrama de flujo: Estado “Charging”.

Finalmente en el estado “Communicating” es aquel en el que se habilita el hotspot y se levanta el servidor de Node-red, aquí se comunica el nido con una computadora en la base del nido, esto se mantendrá hasta que haya pasado el tiempo especificado en

No se deberíamos decirlo en algún lado,
probablemente hardware

. Además se continúan sensando las variables físicas

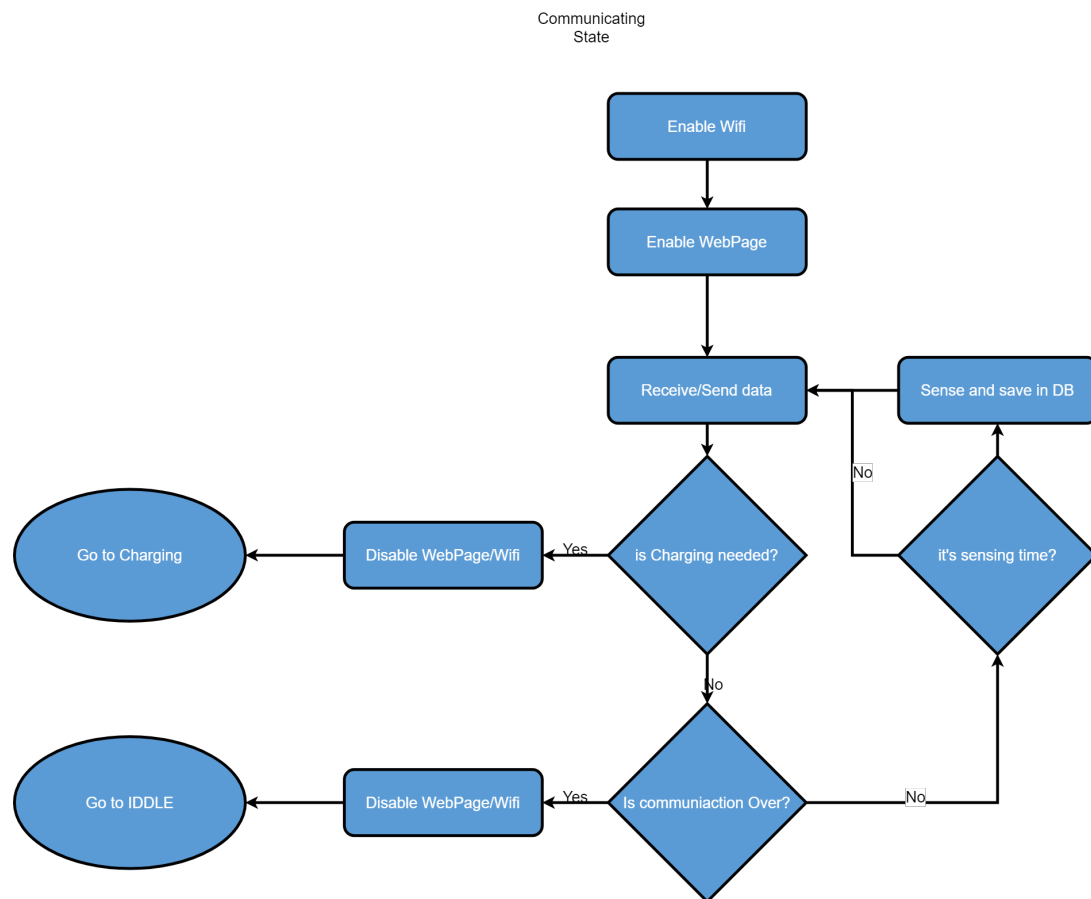
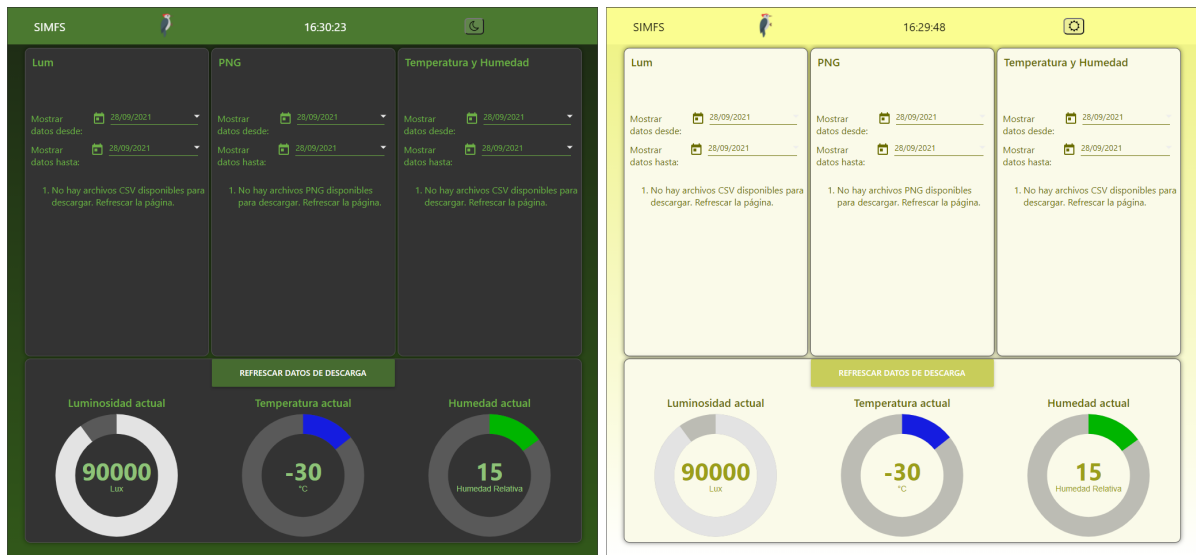


Figura 0.2.2.5: Diagrama de flujo: Estado “Communicating”.

- 0.2.3 Análisis de complejidad
- 0.2.4 Descripción de subrutinas
- 0.2.5 Listado de elementos del código
- 0.2.6 Plan de prueba de módulos y de depuración de Software
- 0.2.7 Front end



(a) Interfaz con usuario versión oscura.

(b) Interfaz con usuario versión clara.

Figura 0.2.7.1: Página del servidor a la cual accede el usuario.