Firmware Integration Documentation

This document lists all the important information needed for a remote firmware team to develop firmware for the smart charger without being in physical possession of the device itself. Any additional material or relevant information if required can be requested and will be added in due course to this document.

GPIO pin mapping

This is the pin mapping for the ESP32-C3-MINI which is the module used in the design.

Function	ESP32 pin	Physical pin	Notes
Charging	IO7	22	Active low GPIO input
Stop charge	IO8	21	GPIO output
I2C SCL	IO0	12	
I2C SDA	IO1	13	

EEPROM connection

The EEPROM is connected to ESP32 over the I2C bus.

EEPROM part number for obtaining datasheet : 24LC512-I/MF

There are 2 EEPROMS connected. I2C Addresses : 0x50, 0x51

ESP32-C3-MINI programming

The ESP32-C3-MINI is configured to be programmed USB CDC and also communicate as a UART COM port overt USB CDC. The code and programming settings on the ESP-IDF has to be configured to do so accordingly.

BLE Service and Characteristics

NAME	UUID	Notes
NUS Service	6E40-0001-B5A3-F393-E0A9-E50E-24DC-CA9E	UART over BLE service
TX Characteristic	6E40-0003-B5A3-F393-E0A9-E50E-24DC-CA9E	Receive data from wearable (can set to notify)
RX Characteristic	6E40-0002-B5A3-F393-E0A9-E50E-24DC-CA9E	Send data to wearable

BLE Commands

These strings are sent over the RX characteristic and a response may be received on the TX characteristic as a result.

1. “reboot\n”
    a. Function: performs a soft reset of the wearable
    b. Returns: N/A
    c. What to implement: N/A
2. “fram,X,Y(,Z)\n”
    a. Function: to read or write the internal FRAM. X can be 1 or 0, 1 for read 0 for write. Y is the address to read or write from it should always be a multiple of 4. If the command is write, then another part Z is included which is the integer value to be written.
    b. Returns: if write nothing is returned. If read the 4 Byte content of the location is retuned as a uint32_t as a string of format “2,Y,Z” where Y is the memory location read and Z is the value at that FRAM location
3. “fram,1,08\n”
    a. Function: using the above format, this returns the time that the device starts timestamping data
    b. Returns: A UNIX time in seconds
4. “data_sync,X\n”
    a. Function: puts the wearable in sync mode and stores current time. The device starts streaming the data stored in its internal memory after this command. X here is the current local time in Unix format. This command should only be called after the command outlined in (3) is called.
    b. Returns: a stream of 128 Byte (this can change in later implementations) BLE packets where the data is stored as bytes. Each sample of data consists of 8 bytes. The 1st 4 bytes are the timestamp, and the 2nd 4 bytes are the sensor values. The timestamp must be added to the starting time from function (3) to acquire the real timestamp. The starting time is formatted in seconds so it must be multiplied by 1000 first to get the millisecond starting time. Each sample can be read as follows:
        i. Time
            1. The 1st byte multiplied by 100 is the millisecond value
            2. The 2nd byte is the second value
            3. The 3rd byte is the minute value
            4. The 4th byte is the hour value
        ii. Sensor values
            1. The 1st 2 bytes represent the first sensor value in little endian
            2. The 2nd 2 bytes represent the second sensor value in little endian
    c. What to implement: The data needs to be parse according to the above scheme and then stored in the external EEPROM as well as transmitted to the backend server.
5. “battery\n”
    a. Function: returns the battery voltage of wearable in ADC units.
    b. Returns: a string of format “0,X” where X is an integer.
    c. What to implement: When the wearable is being charged the system should stop charging and poll the battery voltage at a fixed interval of once every 3 min. The actual voltage is obtained by the following equation – voltage = (X/4095)*7.2

If this voltage value is greater than 4.3 then charging should remain stopped. If voltage is below 4.3 charging should be resumed.

When a connection is made to the wearable it transmits a string every ~500ms of the format:

“1,A,B,C,D,E,F,G,H”

No parsing needs to be done on this string. This is for debug purposes and is sending files to log over BLE on our debugging setup.