Esperto Watch

Hardware Design Description (HDD)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Description | By |
| 2.1 | 11-09-2018 | Version 2 Initial draft | Daniel De Sousa |
|  |  |  |  |

# Project Introduction

The Esperto Watch is a wearable platform in the form of a smart watch which is powerful enough for researchers and developers to use in their professional work yet simple enough for beginners to learn with. The platform is equipped with numerous sensors to detect biometric data such as a user’s heart rate or step count, wireless communication, and a companion mobile and web application to track user metrics.

The watch firmware is fully customizable, allowing developers to build onto our custom algorithms, build their own, or add features such as GPS, activity tracking, or Wi-Fi.

**Figure 1:** The Esperto Watch

# System Overview

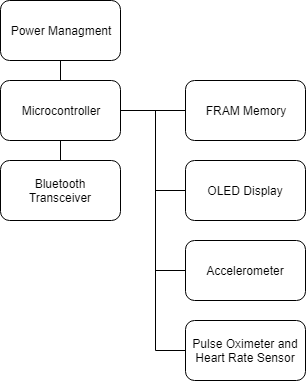
## Feature Overview

The Esperto Watch supports the following features:

* Time tracking, heart rate detection, step detection, SpO2 calculation
* Firmware upgrades / debugging over micro USB (also available over SWD)
* Battery charging
* FRAM user data storage (non-volatile)
* Bluetooth Low Energy communication for use with a mobile application
* Data synchronization between device and mobile application (heart rate, step count, SpO2)
* Mobile notifications (texts, calls)
* Displaying time, device status, mobile notifications, heart rate, and number of steps taken

## Block Diagram

These features are supported by implementing specialized sensors in the hardware and developing algorithms to obtain data from these sensors, filter the data, and convert the raw data into user metrics. A general system block diagram can be seen in **Figure 2.**



**Figure 2:** Esperto Watch General System Diagram

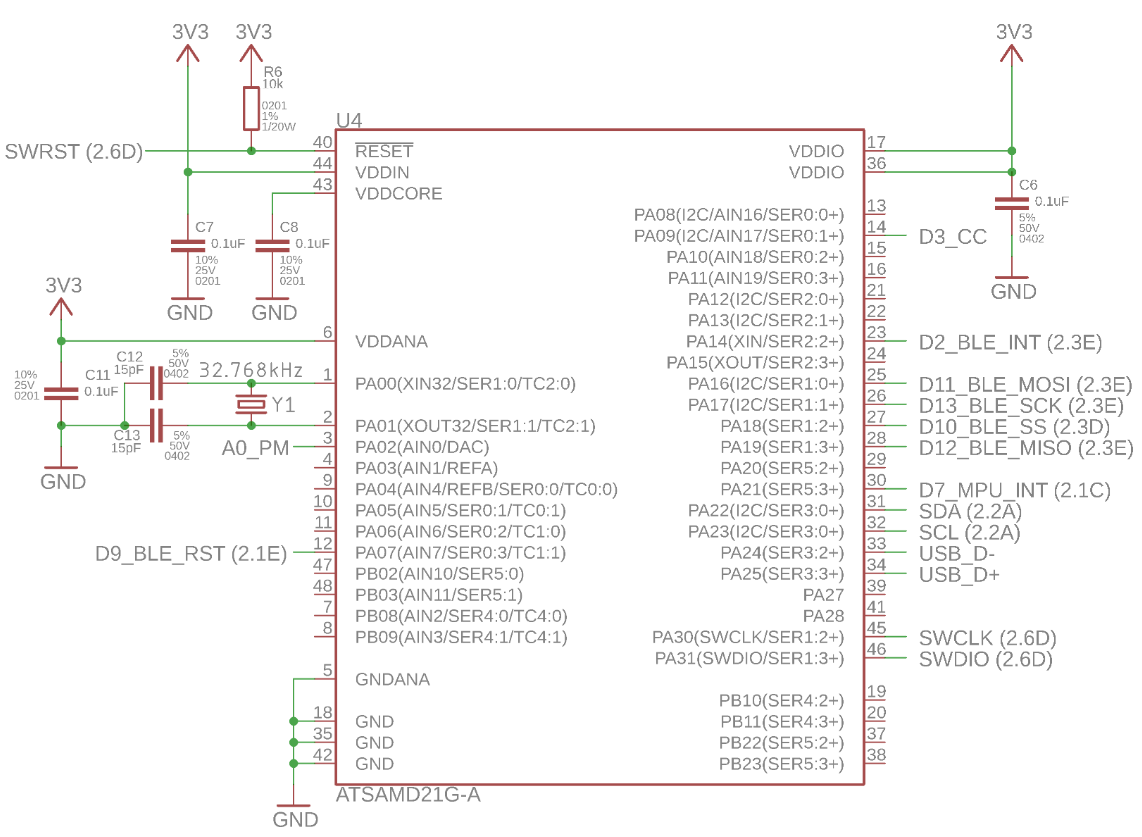
# Hardware

## Microcontroller

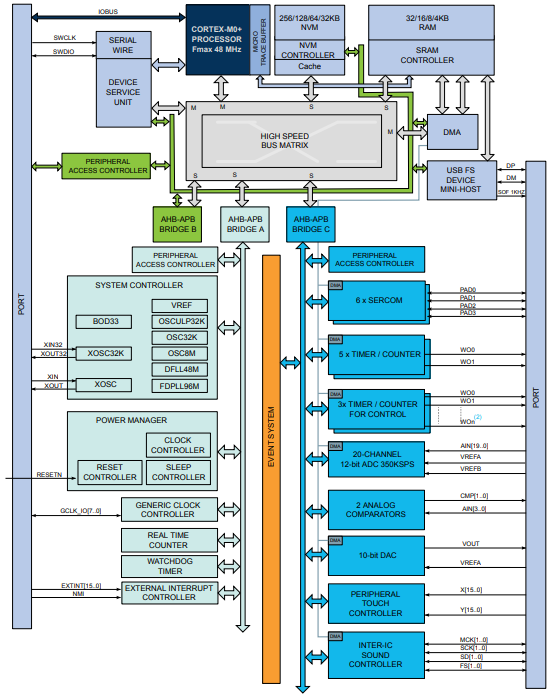
The Esperto Watch microcontroller is the SAMD21G18 by Atmel. The SAMD21 is a ARM Cortex M0+ flash-based controller running at a CPU speed of 48 MHz. It has the following features:

* 256 kB of embedded flash and 32 kB of SRAM
* Low power consumption at less than 70 uA/MHz
* 6 serial communication modules (SERCOM) supporting interfaces including SPI, USART, I2C
* Embedded full speed USB device and host capable of up to 12 Mbps throughput
* 14 12-bit, 350ksps ADC channels and 1 10-bit DAC channel
* Internal 32-bit Real Time Clock (RTC) with calendar functionality
* 256 channel Peripheral Touch Controller (PTC)
* 12 Direct Memory Access Controllers (DMAC)
* CRC-32 generator
* Watchdog Timer (WDT)

The Esperto Watch schematic for the microcontroller can be found in **Figure 1** whereas a system diagram block diagram of the SAMD21 can be seen in **Figure 2.**



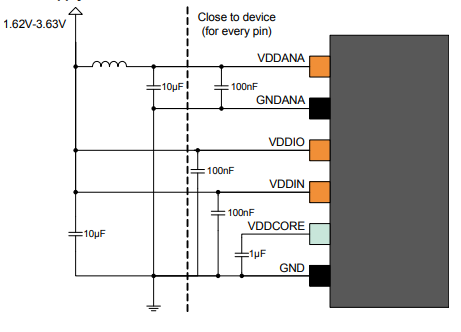
**Figure 1:** Microcontroller Schematic



**Figure 2:** SAMD21 Block Diagram

### Power Supply

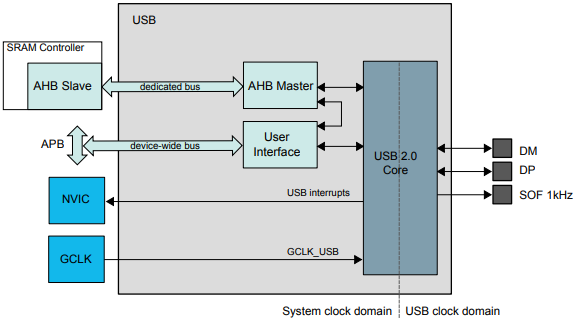
The SAMD21 uses one single main supply ranging from 1.62 V to 3.63 V. The recommended schematic is seen in **Figure 3.** The capacitor connected to VDDIN is equivalent to C6, C11 and C7 connected to the supply input, the IO input, and analog input in **Figure 1.** Their purpose is to decouple high frequency noise between the output of the regulator and the input of the microcontroller. Similarly, the capacitor connected to VDDCORE is used for the internal regulator and is equivalent to C8 on the schematic.



**Figure 3:** SAMD21 Supply Schematic

### Programming and Debugging

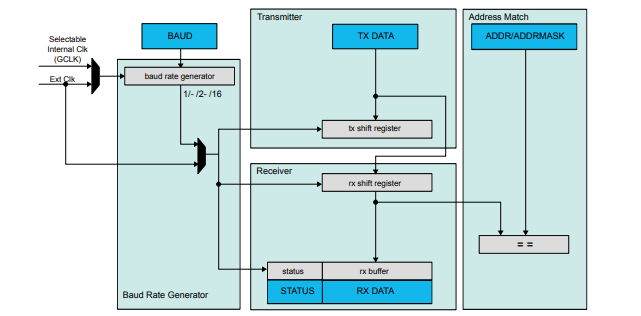
The SAMD21 uses SWD (Serial Wire Debug) to load firmware and debug the microcontroller. This interface consists of a voltage detection line used to determine the logic level, SWDIO or the data line, SWCLK or the clock line, a SWRST or reset line, and a ground reference. However, a bootloader has also been written for the Esperto Watch application which allows the same functionality over the USB interface. This is possible as the SAMD21 can act as a USB 2.0 Host Device. A block diagram of the internal USB interface can be seen in **Figure 4.**



**Figure 4:** SAMD21 USB Interface

### Serial Communication

The SAMD21 has 6 instances of the SERCOM or Serial Communication peripheral. It can be used to support an I2C, SPI, or USART bus. A single peripheral consists of a BAUD rate generator, transmitter, receiver, and an address select. The full SERCOM engine can be found in **Figure 5.**



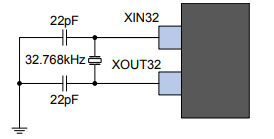
**Figure 5:** SAMD21 Serial Communication Multiplexing Engine

The Esperto Watch uses the following interfaces:

* 100 KHz I2C to communicate with the OLED display, pulse oximeter FRAM, and accelerometer
* 1 MHz SPI to communicate with the BLE transceiver

### Clocks

An external 32.768 KHz crystal oscillator is used as the clock source for the internal RTC or Real Time Clock while other peripherals use the internal 8 MHz oscillator. To use this crystal, 2 load capacitors are required as seen in **Figure 6.**



**Figure 6:** SAMD21 External Crystal Oscillator

The capacitors in this image are equivalent to C12 and C13 in **Figure 1**. In addition to the crystal and internal circuitry of the microcontroller, the capacitors form an oscillator circuit to resonate with the crystal. This causes the crystal to oscillate on its fundamental resonance mode.

### Pinout

The following is the complete pinout of the microcontroller:

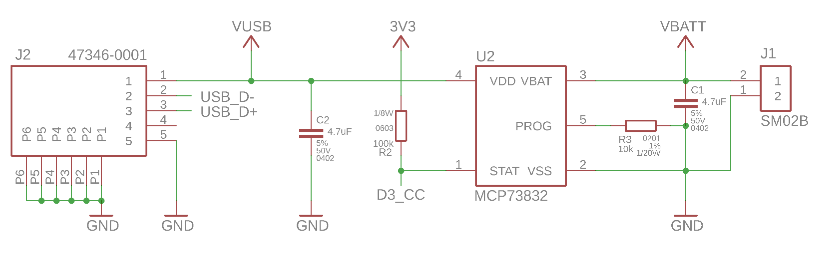
* Digital Pins
  + D2 – BLE interrupt
  + D3 – Charge complete – connected to MCP73832
  + D7 – Accelerometer interrupt
  + D9 – BLE reset
  + D10 – BLE SPI Slave Select
  + D11 – SPI MOSI
  + D12 – SPI MISO
  + D13 – SPI SCK
  + SDA – I2C Data line
  + SCL – I2C Clock line
  + USB D-
  + USB D+
  + SWCLK
  + SWDIO
* Analog
  + A0 – Connected to power monitoring circuit

## Power Management

Power management circuitry is used to regulate the internal supply to a voltage level suitable for the microcontroller and all the peripherals, charge the battery, and also switch between USB and battery.

### Battery Management

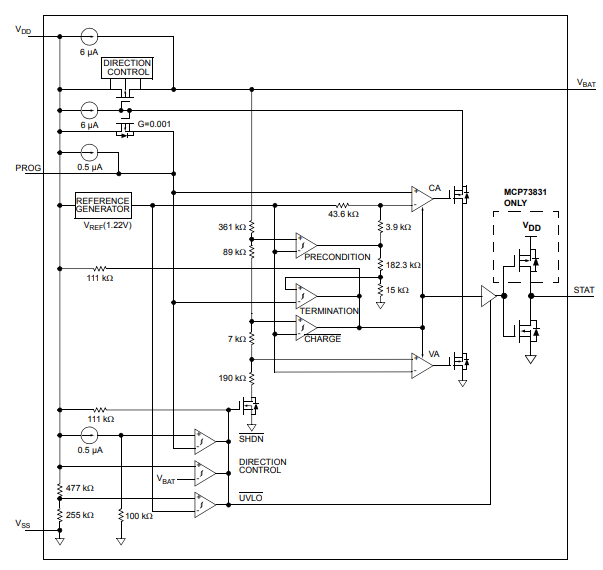
The MCP73832 lithium polymer charge management controller is responsible for charging the rechargeable lithium polymer battery when USB is plugged in and notifying the microcontroller of the battery state. The schematic can be seen in **Figure 7.**



**Figure 7:** Battery Management Schematic

5 V supply is provided by the USB connector, annotated by J2, to the MCP73832. The signal is bypassed by capacitors, annotated by C2 and C1 on the input supply and battery supply. PROG or the current regulation pin is tied to ground reference by a resistor. The value of this resistor determines the charging rate using the following formula:

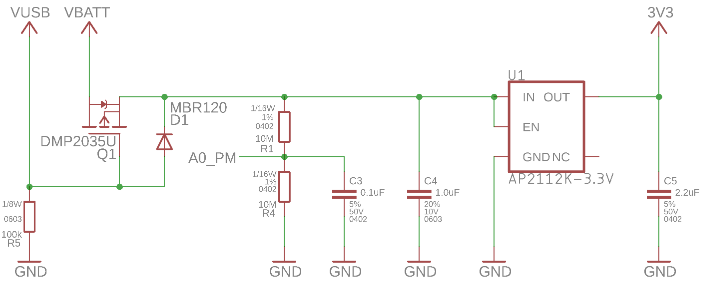
In the previous figure, R3 is the programming resistor which determines the charging rate. The value was chosen to be 10 kΩ and thus, the charging current is 100 mA. This charging current is the current recommended by the battery manufacturer. Finally, the STAT pin is used to notify the microcontroller when the battery is finished charging. In the MCP73832, this open-drain pin is driven low when the battery is fully charged. Otherwise, the line is high impedance and will be pulled up by R2. This line is read by one of the microcontrollers digital pins. This driver can be seen in the MCP73832 block diagram in the following figure.



**Figure 8:** MCP73831/2 Block Diagram

### Switch Supply and Low Dropout Regulator

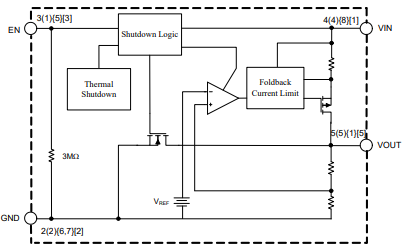
The DMP2035-U P-Channel MOSFET and AP2112 LDO or Low Dropout Regulator are used to switch between USB and battery supply quickly and regulate it down to 3.3V as seen in **Figure 9.**



**Figure 9:** Switch Supply and Linear Regulator Schematic

As mentioned, the MOSFET is used to quickly switch between the 2 supplies. When the gate of the FET is low or below a certain threshold voltage, the USB is not plugged and thus, the drain and source are connected, allowing the battery to supply the device. However, when USB is plugged, the FET will be turned off and USB will power the device. R5 is used to pull down the USB supply line to ensure it is logic low when USB is not plugged in. The MBR120 Schottky diode is used to prevent reverse voltage while maintaining a fast switching speed. R1 and R4 are used in a voltage divider configuration to divide the input voltage, whether battery or USB, in half. This is required as the microcontroller analog pins are 3.3 V tolerant and any voltage greater may cause damage. This analog pin is used to determine the state of the supply and whether USB is plugged in and how charged the battery is. However, to prevent a large constant current draw, the resistors have large values of 10 MΩ. Since these values are large, they may not be able to provide the ADC or Analog to Digital Converter with enough current and thus, C3 is used to store small amounts of charge.

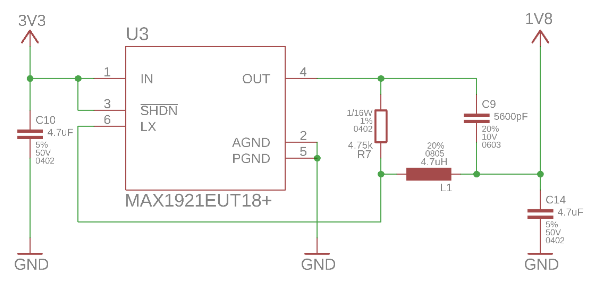
Finally, the API2112k-3.3 LDO is used to regulate a voltage up to 6.5 V down to 3.3 V. C4 and C5 are used as decoupling capacitors. The enable pin is tied to the supply to ensure that the LDO is constantly enabled. A block diagram of the AP2112 can be seen in the following figure, outlining its internal circuitry.



**Figure 10:** AP2112 Linear Regulator Block Diagram

### Buck Regulator

The MAX1921-18 low voltage buck regulator is used to convert 3.3 V outputted by the LDO to 1.8 V. 1.8 V is required by the MAX30102 pulse oximeter while the rest of the devices operate at 3.3 V. A buck regulator is used as it is much more efficient than an LDO at large input-output voltage drops, achieving up to 95% efficiency. **Figure 11** outlines the schematic for the MAX1921 buck regulator.

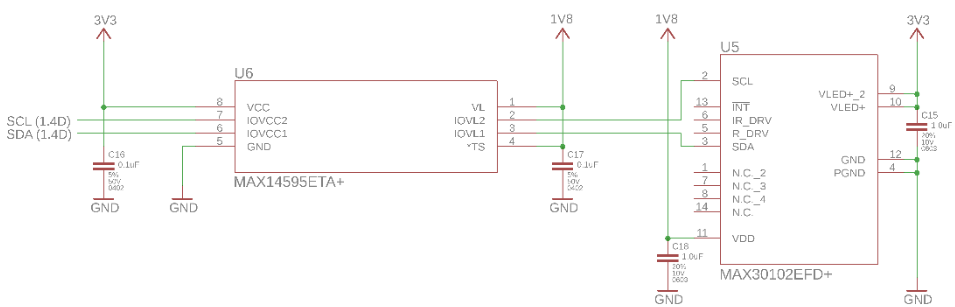


**Figure 11:** Buck Regulator Schematic

Buck regulators are a form of DC-DC step-down converter which switch rapidly, in this case at 1.2 MHz, to produce a lower fixed voltage from the input. C10 and C14 are connected to the input and output supply of the buck regulator to filter out any ripple currents or voltage peaks and drops. The shutdown pin is connected to the supply so that the regulator always remains enabled. The LX pin is connected to an inductor and is used to collect feedback from the output circuitry. R7 is used to select which factory configured output voltage is required while C9 acts as a feed-forward capacitor to ensure stability in the system.

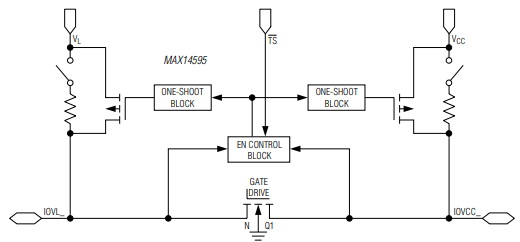
## Pulse Oximeter and Heart Rate Sensor

The MAX30102 pulse oximeter is used to obtain infrared and red-light readings in order to determine a user’s heart rate and blood oxygen concentration. The MAX14595 low-power dual-channel logic-level translator is used to interface the SAMD21’s 3.3 V logic with the MAX30102’s 1.8 V I2C logic. A schematic can be seen in the following figure.



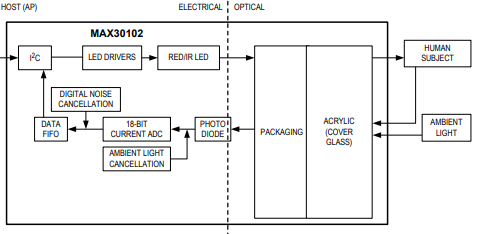
**Figure 12:** Pulse Oximeter Sensor Schematic

As mentioned, the role of the MAX14595 is to shift the signals’ logic level in a bidirectional manner. C16 and C17 act as bypass capacitors before the level translator for both logic supplies. This translator can support speeds up to 8 MHz. However, this application only uses an I2C bus clocked at 100 KHz. The TS or shutdown pin is constantly driven high to ensure constant normal device operation. I2C pull-ups are not required for the I2C bus as the level shifter has internal pull-ups which can pull-up the entire bus as seen in the block diagram below. These resistors have a typical value of 7 kΩ.



**Figure 13:** MAX14595 Level Shifter Block Diagram

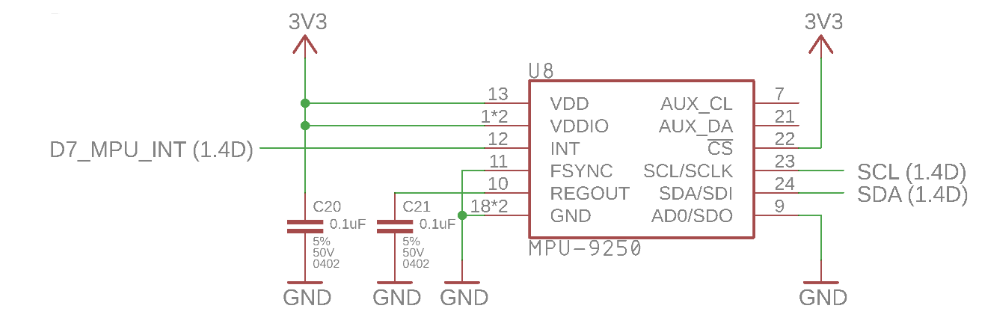
As mentioned, the optical sensor operates on 1.8 V and communicates with the SAMD21 over the I2C interface. The MAX30102 is also supplied with 3.3 V for the red and infrared LEDs. All supplies are connected to bypass capacitors, C15 and C18. A block diagram demonstrating the internal ADC, serial interfaces, LED drivers, and filters can be seen below.



**Figure 14:** MAX30102 Block Diagram

## Accelerometer and Gyroscope

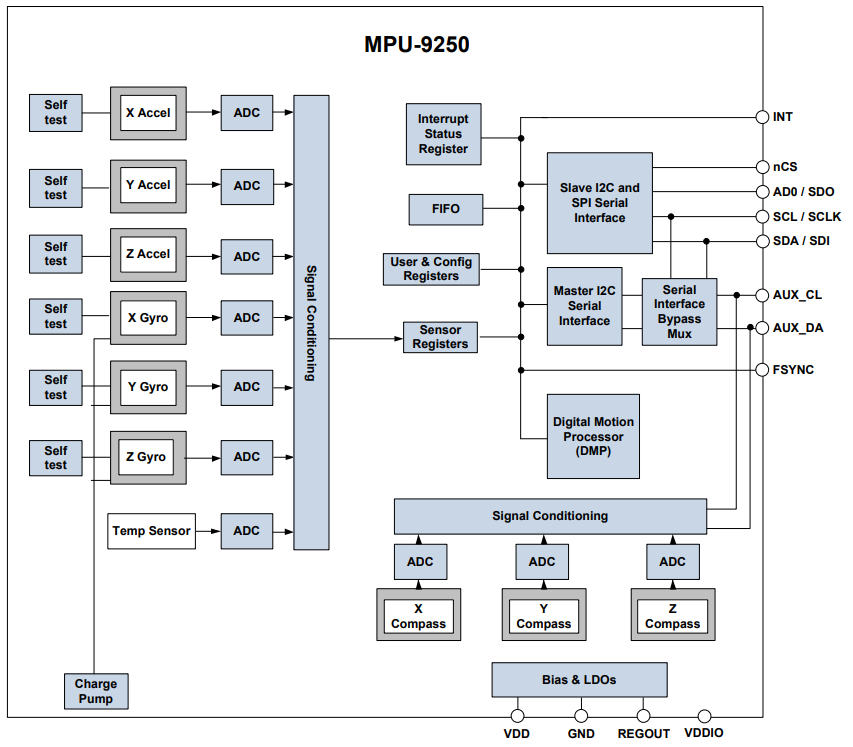
The MPU-9250 9-axis motion tracking device is used to detect steps, user movement, and used for motion wakeup detection. The MPU-9250 houses a 3-axis accelerometer, gyroscope, and magnetometer. The schematic can be seen in the following figure where C20 acts as a bypass capacitor for the input supply and C21 acts as a filter capacitor for the internal linear regulator.



**Figure 15: Accelerometer Schematic**

The MPU-9250 has both an I2C and SPI interface. This implementation uses the 100 KHz I2C bus to communicate with the SAMD21. Furthermore, there is also an interrupt pin which connects to a digital pin on the SAMD21 which is active when a motion event occurs. Furthermore, the AD0 pin is tied to ground to select the base I2C address. This pin can be tied high to add 1 to the base I2C address.

Internally, the MPU-9250 consists of the serial interface, the DMP or Digital Motion Processor which performs some of the data filtering and processing, data and configuration registers, an LDO regulator, a charge pump to generate the high voltage required for the MEMS oscillators, and the sensors themselves as seen in the following block diagram.

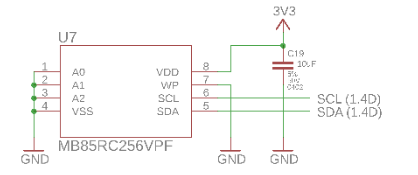


**Figure 16**: MPU-9250 Block Diagram

## Non-Volatile Memory

For user data storage, a MB85RC256V 32 kB FRAM chip is used over the I2C interface. FRAM or Ferroelectric Random-Access Memory is a memory storage like SRAM as it is not prone to wear leveling over time. However, unlike SRAM, it retains memory even when power is not being supplied

The MB85RC256V operates at a frequency of 1 MHz and on a 100 KHz I2C bus. The schematic can be seen in the following figure where C19 acts as a bypass and bulk capacitor. The WP or write protect line is held low to allow for continuous writing.

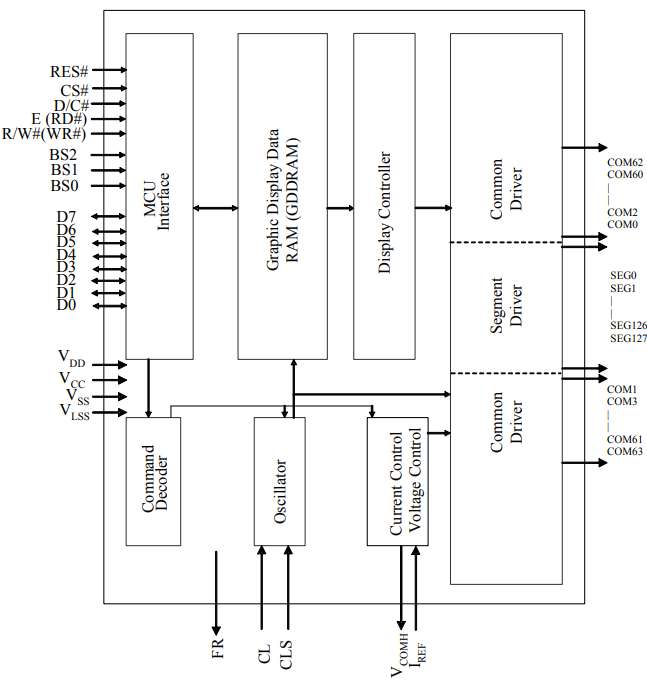


**Figure 17:** FRAM Schematic

Additionally, the address lines, A2, A1, and A0 are all tied to ground to assign the memory its base I2C address. These pins can be used to add a 3-bit number to the base address of the chip.

## OLED Display Driver

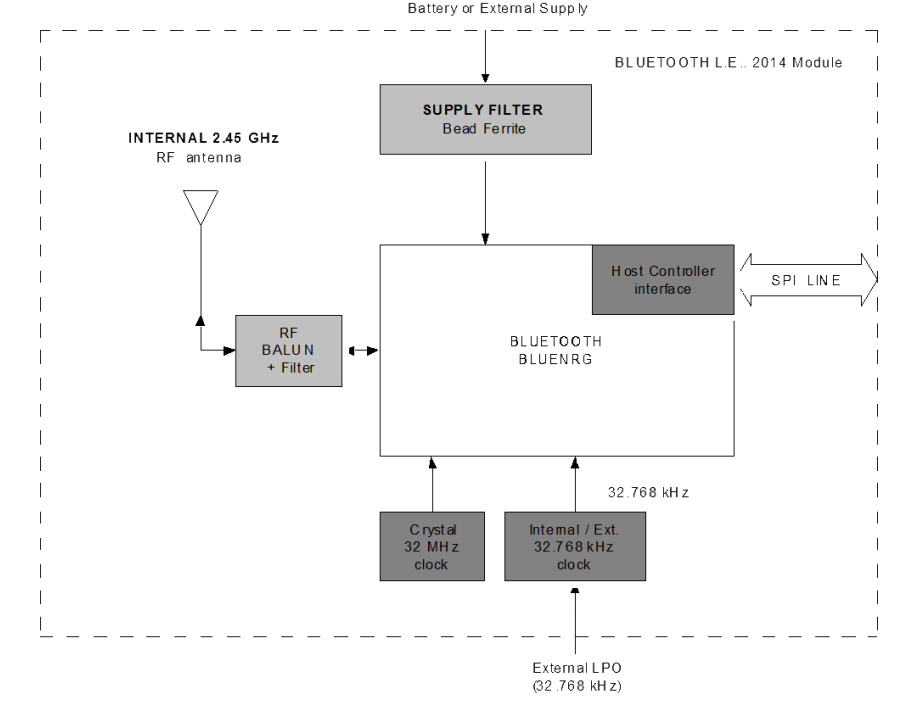
The SSD1306 OLED display driver is used to drive a 128 x 64 pixel dot-matrix OLED display. It communicates with the microcontroller over the I2C interface. The internals of the driver consist of the serial interface, graphical display data storage, and a display driver which consists of a segment and common driver as seen in the following figure.



**Figure 18:** SSD1306 Block Diagram

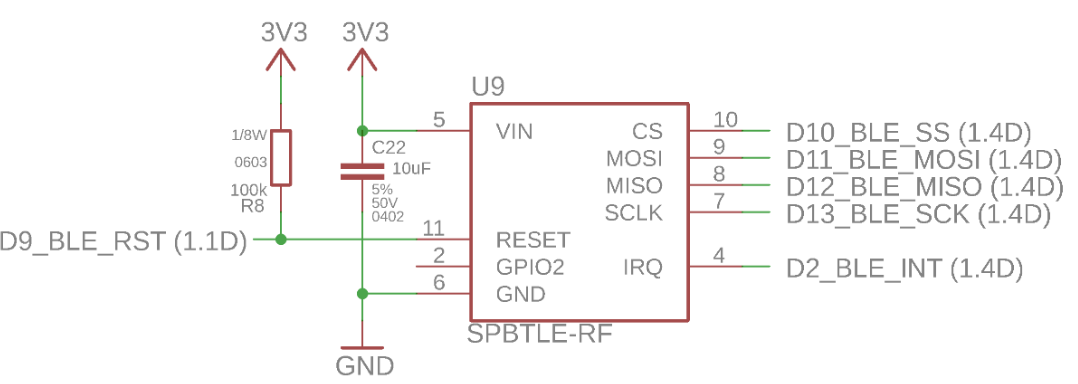
## Bluetooth Transceiver

The SPBTLE-RF low-power Bluetooth Low Energy or BLE network processor is used to communicate with mobile devices over BLE. This module communicates with the SAMD21 over the SPI interface and contains the hardware essentials to enable wireless connectivity including the external oscillators, the RF balun acting as the antenna, and host controller interface as seen in the following figure.



**Figure 19:** SPBTLE-RF Block Diagram

The SPI interface requires a transmit and receive data line, a clock line, and a chip select line. Additionally, a reset line is connected to a digital pin on the SAMD21 to reset the host controller interface when the SAMD21 boots. An interrupt line is connected from the modules IRQ line to the microcontroller and is set when a message has been received from a mobile device. A schematic of the module can be seen in the following figure where R8 is used to pull-up the reset line and C22 is used as a bulk capacitor for the supply.



**Figure 20:** SPBTLE-RF Schematic

# Notes