

# 电气系统

## SteamVR™ Tracking

### 简介

SteamVR™ Tracking 电气系统囊括了实现姿势定位所需的所有功能以及控制器和 HMD 的基本外设要求。电气系统的基本要求是能够检测最多 32 个光学传感器并加上时间戳，这些传感器将接收定位器发出的参考信号。一旦生成了时间戳，可使用 USB 或 2.4 GHz 无线链接将其传输到 SteamVR™ 计算机。除了传感器，还将生成一串 IMU 数据并传输到计算机，将其记录下来。除此核心功能以外，还实施了控制器和 HMD 的一些基本功能。

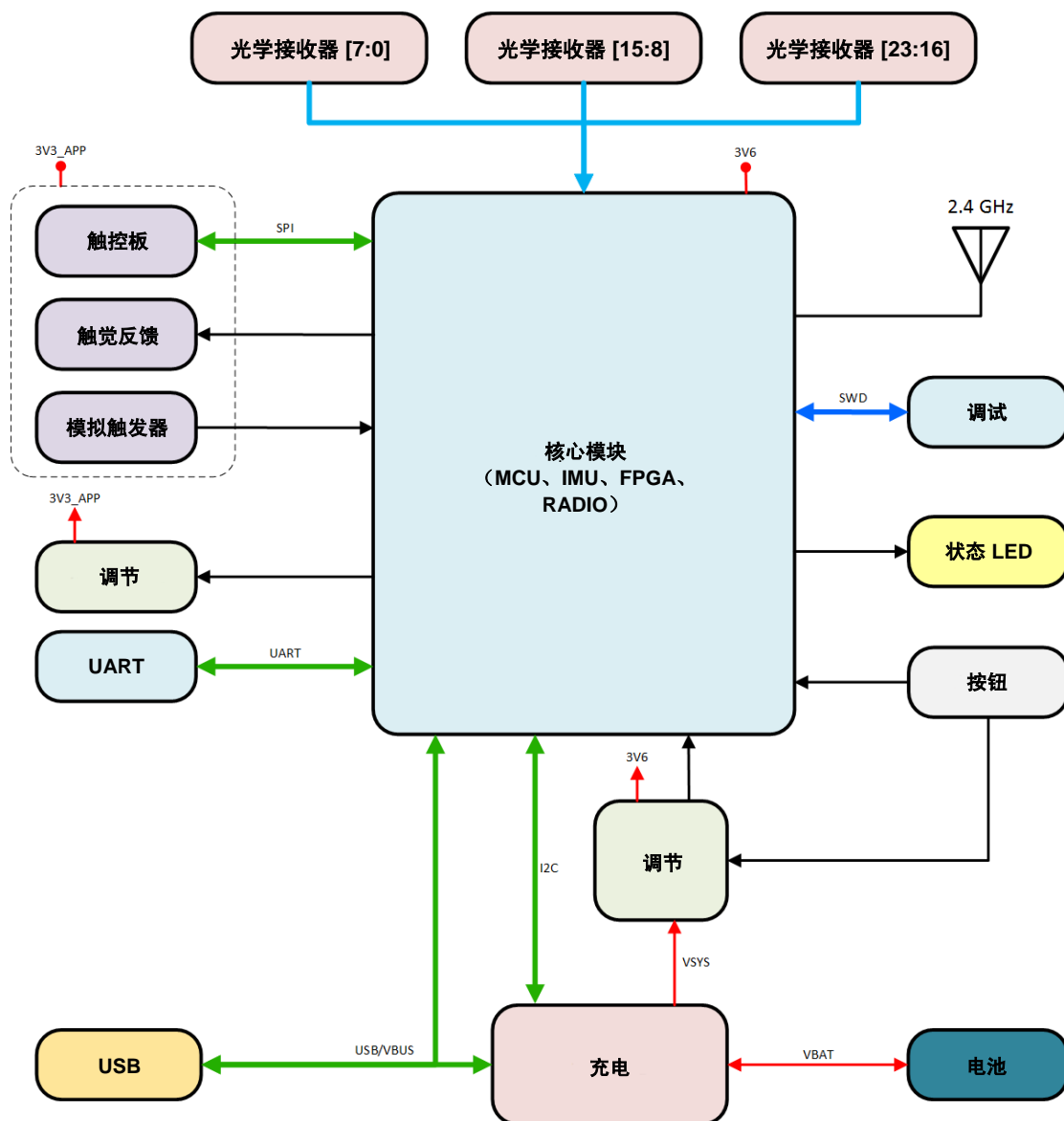
控制器和 HMD 具有不同的外围设备要求，其中有一部分被集成到 SteamVR™ Tracking 电气系统之中。控制器需要电池电源和用户控件，例如按钮、触控板、LED 和触觉反馈。电气系统包含电池充电和电量计电路，以管理锂离子电池组。它还包含基本控件，包括 RGB 状态 LED、按钮、磁性控件的霍尔效应传感器、电容式触控板和触觉反馈。通过系统微控制器上灵活的输入和输出外围设备，我们还可将这些控件替换为各种其他控件和外围设备。HMD 需要一个串行闪存以存储显示校准数据。尽管 HMD 没有那么多的用户可访问输入和输出，但这些外围设备可用显示桥和其他视频外围设备替换。

本文档将介绍 SteamVR™ Tracking 参考设计的电气架构，并着重给出了设计和使用方面的建议。本文档中的零件号和信号名称均与 SteamVR™ Tracking HDK 中的参考设计示意图相对应。

## 框图

### 控制器

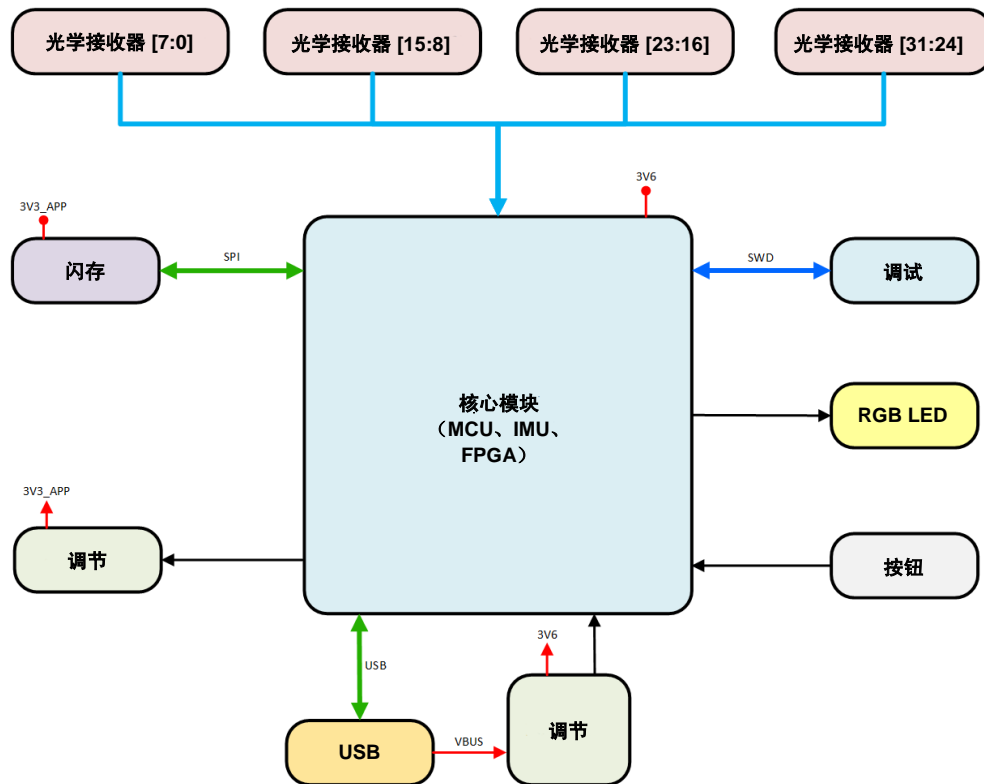
下图显示控制器的高级框图。



因尺寸所限，控制器包含的传感器通常少于 32 个。同时，控制器通常支持无线操作。无线操作要求核心模块具有无线电和天线，以进行无线通信。无线操作还意味着需要电池电源，这意味着需要增加一套电池、充电电路和电量计。电池通过 USB 接口充电。控制器还需要用户输入和输出外围设备。SteamVR™ Tracking 参考设计支持的输入控件包括五个按钮、一个电容式触控板和一个用于模拟触发器的霍尔效应传感器。控制器通过 RGB 状态 LED 和触觉反馈传达使用反馈。控制器通过 USB 或 2.4 Ghz 无线链接与 SteamVR™ 计算机通信。最后，有多个通信接口用于支持系统开发和制造，包括用于无线电和微控制器的单线调试端口，以及用于调试和制造测试的 UART 连接。此框图并未囊括所有可设计到控制器中的可能功能集，但已显示了应用程序开发人员目前在创建 SteamVR™ 应用程序时所用的一组基本控件。

## HMD

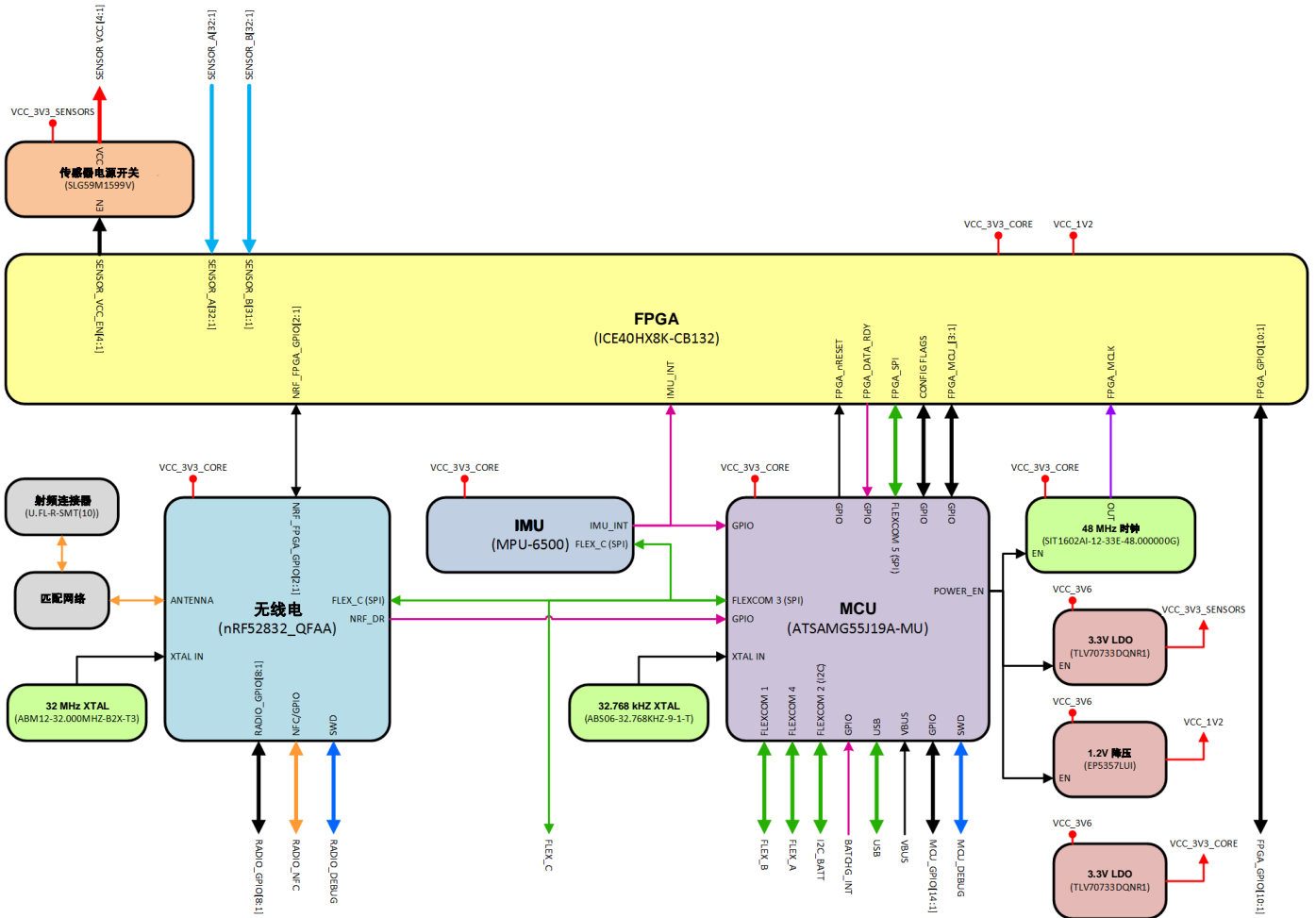
SteamVR™ 的高级框图



HMD 由于尺寸较大加上必须要获得极为可靠的定位性能，因此通常会包含近 32 个传感器。与控制器不同，HMD 不需要无线操作，因此就无需无线电、电池以及相关电路。通常，HMD 上的用户控件也会缩减至一个状态 LED 和一个按钮。HMD 通过 USB 与 SteamVR™ 计算机通信。就像控制器一样，HMD 也有多个通信接口用于支持系统开发和制造，包括用于微控制器的单线调试端口，以及用于调试和制造测试的 UART 连接。此框图显示了定位系统的要求，但略去了视频流和显示所需的外围设备。HMD 的具体要求决定了这些外围设备，可通过定位核心模块中的 MCU 或系统中另外加设的 MCU 来控制这些外围设备。

## 核心功能

核心功能框图如下所示。



FPGA 将为来自于光学传感器的信号以及由 IMU 使用 48 Mhz 时钟驱动的捕捉寄存器获取的测量结果加上时间戳。这些时间戳被 MCU 读取，并通过无线电或 USB 接口传输至 SteamVR™。请注意，只有无线设备（例如控制器）才需要无线电。

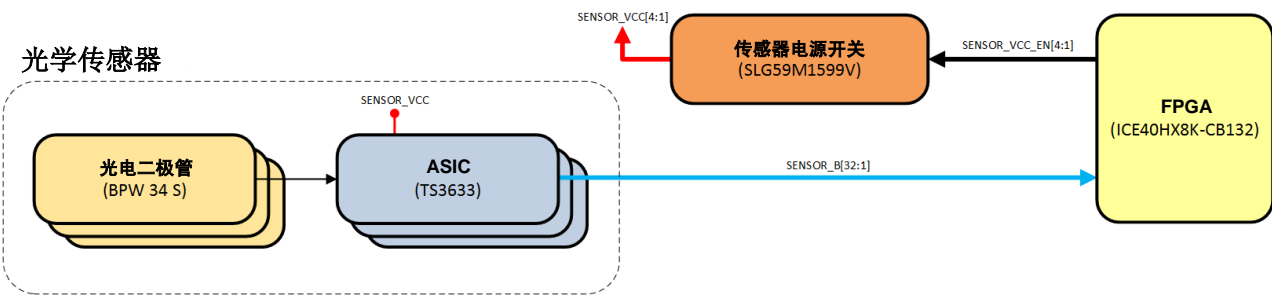
核心内包括多个稳压器，用于为 MCU、无线电、FPGA、IMU 和光学传感器生成所需电源轨。MCU 能够关闭光学传感器和 FPGA 核心的电源。当 FPGA 核心的电源被关闭时，48 Mhz 振荡器也会被禁用。通过 FPGA，MCU 还能控制四个 PFET 开关，从而控制光学传感器的时序电源。

从 MCU 可接出多种总线和 GPIO，以连接用于不同应用的外围设备。FPGA 还提供了 10 个 GPIO 信号，用于应用特定的功能。

# 核心功能

## 光学传感器

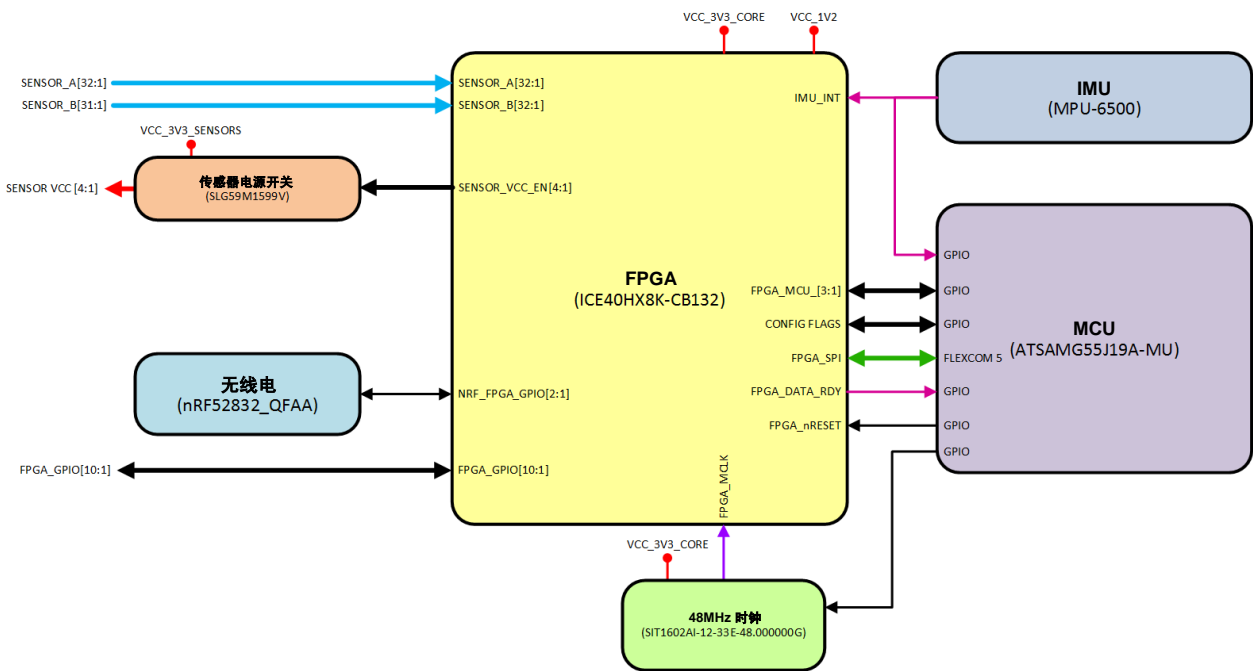
光学传感器包含一个连接到定制 ASIC（特定用途集成电路）的光电二极管，此 ASIC 起到一个互阻抗放大器 (TIA) 和包络检波器的作用。光电二极管将入射红外光转化成电流。然后该电流又通过 ASIC 的 TIA 转换成电压。参考信号中的调制频率在 ASIC 中被滤除，所产生的包络被传输到 FPGA 以加上时间戳。



系统最多可支持 32 个光学传感器。它们通过四个 3.3V 电源轨中的其中一个接收电源，即 SENSOR\_VCC[4:1]。光学传感器通过 SENSOR\_B[32:1] 将入射红外光的包络传输到 FPGA。

## FPGA

FPGA 的主要功能是对 48 Mhz 系统时钟上的光电探测器和 IMU 信号加盖时间戳。



传感器输出以 SENSOR\_B[32:1] 信号到达 FPGA。信号 SENSOR\_A[32:1] 留待后用。IMU 中断均通过信号 IMU\_INT 加盖时间戳。

FPGA 中的 32 位捕捉寄存器由 48 Mhz 时钟驱动，该时钟则由 FPGA\_MCLK 传递。FPGA 还使用 SENSOR\_VCC\_EN[4:1] 控制四个 PFET 开关，从而控制光学接收器的时序电源。

FPGA 可提供的 10 个 GPIO 信号通过 FPGA\_GPIO[10:1] 提供，以连接用于加盖时间戳的辅助信号、扩展 GPIO 或实施其他功能。

MCU 与 FPGA 在 SPI 上通信，以接收时间戳信息并在系统重置后配置 FPGA。

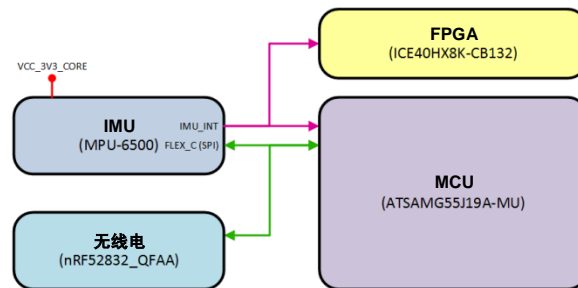
FPGA 与无线电之间的两个 GPIO 信号 NRF\_FPGA\_GPIO[2:1] 留待后用。

## 设计说明

FPGA 的电源包括用于 I/O 环的 3.3V 电源和用于核心的 1.2V 电源。尽管 1.2V 电源轨由 MCU 控制，但 3.3V 电源轨保持上电状态，以避免 MCU 和无线电上的已上电 I/O 焊盘引脚与 FPGA 上的未上电 I/O 接通。使用 FPGA GPIO 信号驱动可能连接到其他电源的其他外围设备时，请记住只要系统还接通着电源，FPGA 的 I/O 环就会处于通电状态。

## IMU

IMU 在 HMD 中以 1000 Hz 频率测量定位对象的平移和旋转加速度，在控制器中则以 250 Hz 频率执行此测量。



MCU 使用 SPI 总线 FLEX\_C 读取测量值。IMU 还与无线电共用 FLEX\_C。

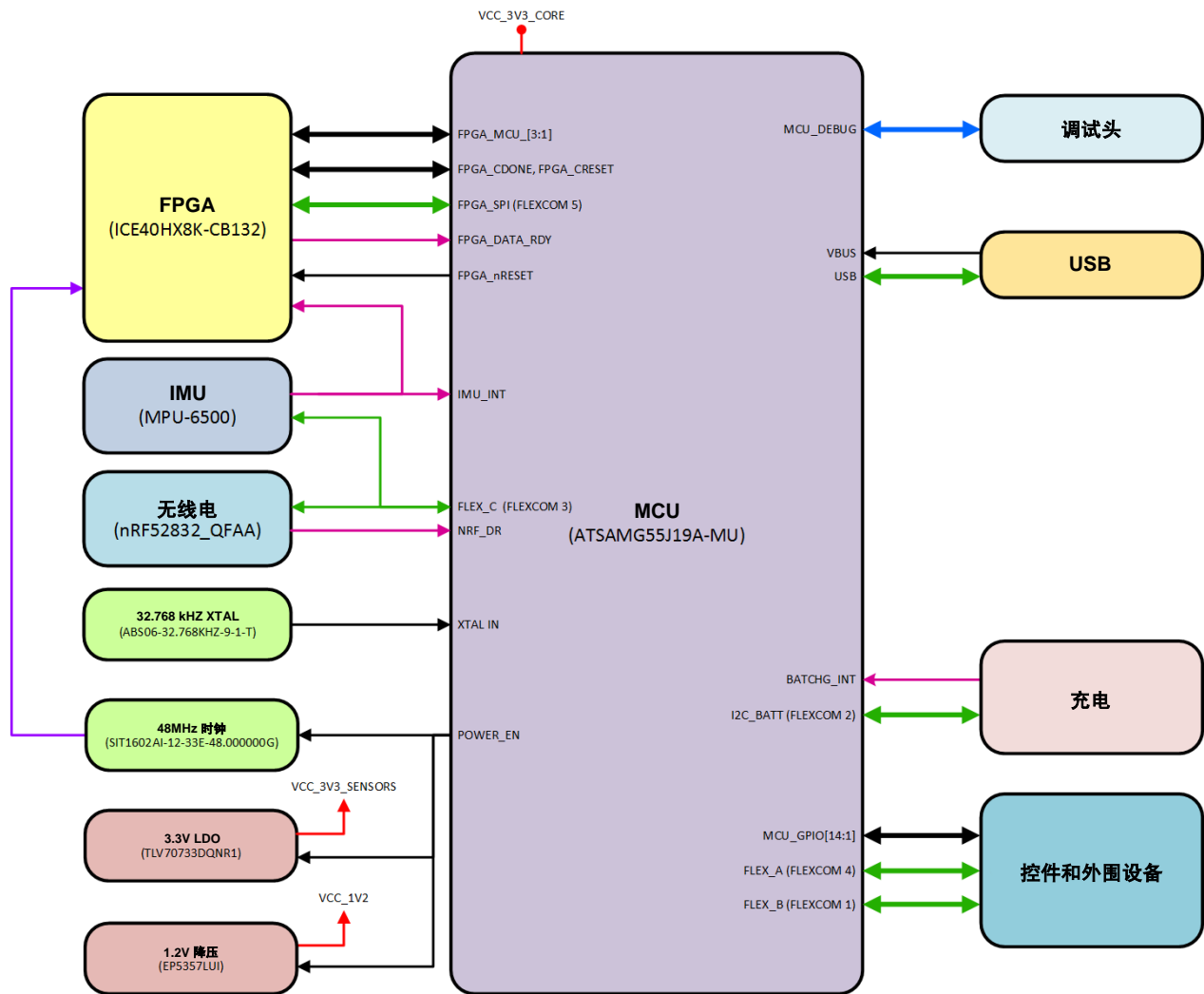
IMU\_INT 信号由 FPGA（用于时间戳）和 MCU（用作数据就绪中断）共用。

## 设计说明

布置 MPU-6500 时，注意在封装的下面布置一个禁止布线铜层。尽管 QFN 封装带有热风焊盘，但不建议将此焊盘连接到 PCB。否则与此焊盘相关的机械应力会影响到测量值。

## MCU

MCU 是电气系统的中央处理器。它负责控制外围设备的电源和启动流程，存储 FPGA 映像并在重置后配置 FPGA，将时间戳从 FPGA 路由到无线电或 USB 接口，通过无线电或 USB 传输 IMU 测量值，管理电池外围设备，以及驱动用户输入和输出外围设备。



MCU 使用同一个 FPGA\_SPI 接口进行配置和通信。进行配置时，MCU 控制 CRESET 并监控 CDONE，同时使用 FPGA\_SPI 发送 FPGA 比特流到设备。配置后，MCU 使用 FPGA\_SPI 控制 FPGA 中的功能，并读取 FPGA 生成的时间戳信息。

MCU 还会使用 SPI 总线 FLEX\_C 与 IMU\_INT，读取 IMU 中的加速数据。无线电与 IMU 共用 FLEX\_C SPI 总线，并使用 NRF\_DR 让 MCU 中断，以进行数据传输。

当系统需要进入较低功率状态时，MCU 可以使用 POWER\_EN 信号关闭传感器和 FPGA 核心的电源。

在电池供电的应用中，MCU 通过连接到 I2C\_BATT 的充电和电量计外围设备管理电池。外围设备可使用 BATCHG\_INT 中断 MCU。

MCU 包括 USB 2.0 Full Speed 外围设备，此外围设备连接到运行 SteamVR™ 的主机上。

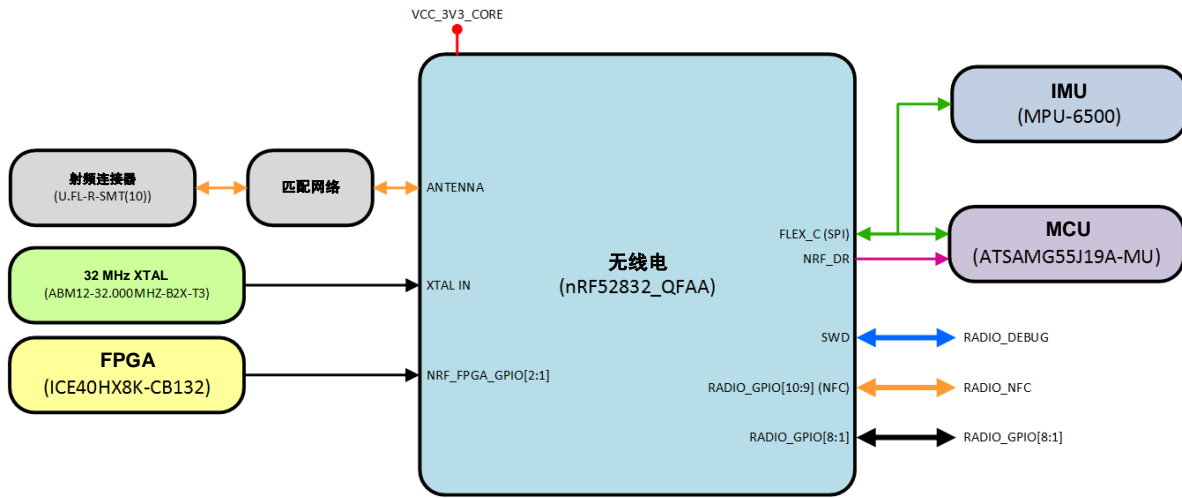
HMD 和控制器的控件和外围设备均连接至 MCU\_GPIO[4:1]、FLEX\_A 和 FLEX\_B。这些引脚可能会通过多种方式进行重新配置，以实施 SPI、I2C、UART 和 GPIO 接口。

## 设计说明

当无需电池充电时，还可将用于管理电池电路的信号收回并用于其他外围设备。

## 无线电

无线电在定位对象与 SteamVR™ 计算机之间提供一个 2.4 GHz, 1 Mb 的无线链接。无线电包括一个自定义固件, 该固件使用 Nordic Enhanced ShockBurst 协议与运行有互补性固件且通过 USB 连接的接收器通信。无线接收器可插入到 SteamVR™ 计算机上的 USB 端口或集线器中, 也可以集成到 HMD 中。



无线电与 MCU 通过 SPI 总线 FLEX\_C 进行通信。当数据准备就绪时, 它使用中断命令行 NRF\_DR 向 MCU 发送信号。

射频天线信号在连接到 2.4 GHz 天线之前会通过一个无源匹配网络。在参考示意图中, 天线连接作为 U.FL 连接器保持开放状态, 因此可选用不同的天线设计和布置方式。

无线电与 FPGA 之间的两个 GPIO 连接 NRF\_FPGA\_GPIO[2:1] 留待后用。

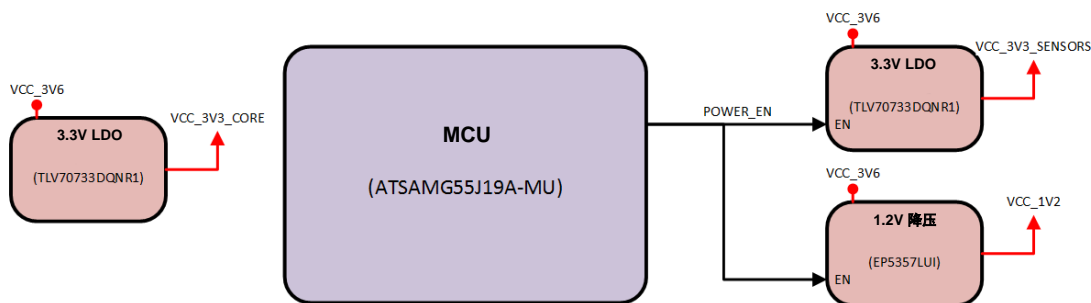
10 个 GPIO 连接 RADIO\_GPIO[10:1] 也留待后用。Nordic NRF52 支持 NFC, 但 SteamVR™ Tracking 不使用 NFC。

## 设计说明

注意遵循数据表中所规定的 Nordic 无线电的布局建议。

## 核心调节

核心中的稳压器为核心功能提供所需的电源电压。



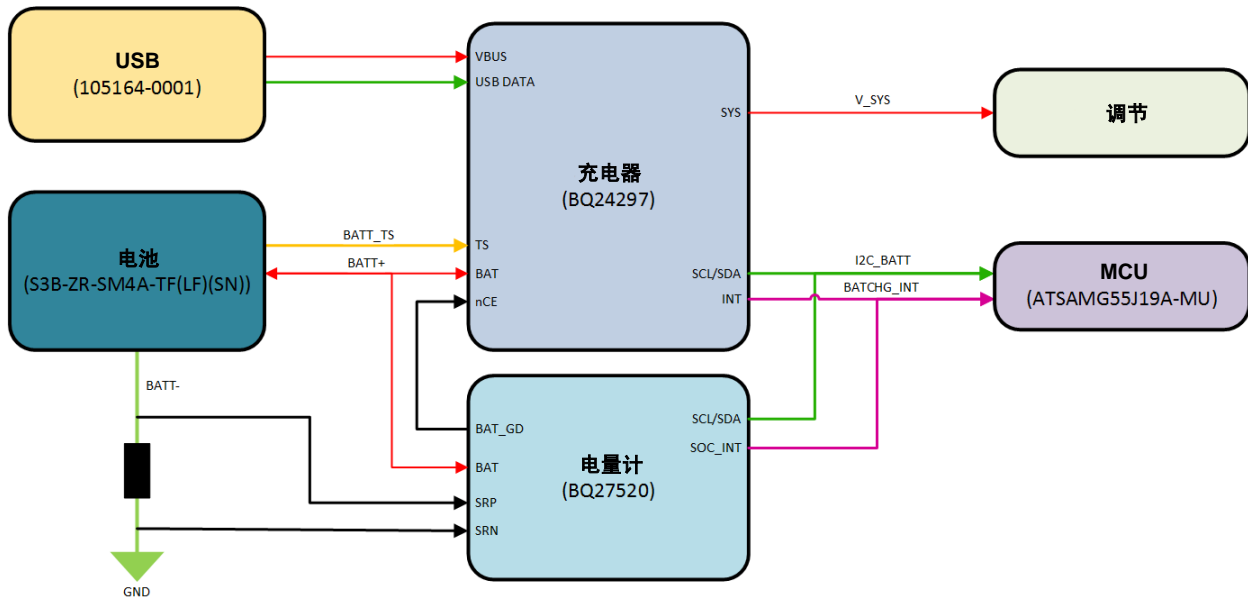


核心稳压器均由 3.6V 系统轨供电。其中一个 3.3V 的 LDO（低压差线性稳压器）构建了 VCC\_3V3\_CORE 轨，为 MCU、IMU、无线电和 FPGA I/O 供电。降压稳压器生成 VCC\_1V2 轨，为 FPGA 核心供电。另一个 3.3V 的 LDO 构建了光学传感器的模拟电源电压，将传感器的灵敏模拟电路与数字噪声相隔离，并在系统进入低功耗状态时允许 MCU 关闭传感器的电源。当 MCU 关闭核心模块的电源时，FPGA 的核心电压也会被禁用。

## 控制器功能

### 充电

充电外围设备管理为控制器供电的锂离子电池。



充电器通过电池或 VBUS 为系统供电。BQ24297 能够一边为电池充电一边为系统供电。充电器产生的系统电压  $V_{SYS}$  已经过调节。但是，充电 IC 会维持  $V_{SYS}$  上的电压，使其足以对电池充电。因此， $V_{SYS}$  上的电压可能介于 3.5V 和 4.35V DC 之间，具体取决于电池状态。为此，需要在下游进行调压。

充电器还能够监控 USB 数据线，以检测连接到系统的主机端口的类型。根据端口类型，充电器自动选择合适的输入电流限值，以遵循 USB 规范。

充电器通过 TS 输入监控电池组中内置的 10k 热敏电阻。如果电池温度超出可接受范围，充电器会停止给电池充电。

电量计 IC 也会监控电池。电量计通过 BATT+ 测量电池电压，通过连接到 SRP 和 SRN 的低端电流检测电阻测量电池电流，从而计算电池中存储的能量。如果电量计检测到电池中有任何问题，它能够通过抬升充电器上的充电使能引脚来禁用充电。

充电器和电量计均通过 I2C 总线和中断信号连接到 MCU。当充电器或电量计请求中断时，MCU 会读取这两者的测量值以确定中断来源以及任何已更改的寄存器值。

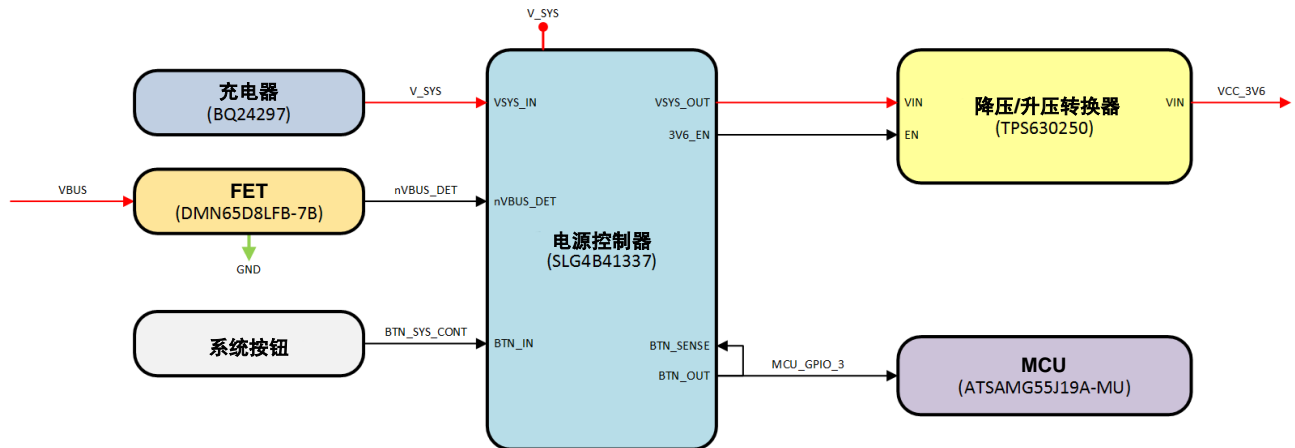
MCU 通过 I2C 总线 (I2C\_BATT) 管理充电器的输入电流限值。根据 USB 主机所赋予的配置，输入电源的限值被设置为在不超出 USB 规范的前提下，尽可能快地给电池充电。

MCU 还使用 I2C 总线 (I2C\_BATT) 从电量计中读取电池电量。

## 调节

### 3.6V 系统电源

调节部分可控制和调节为电气系统供电的 VCC\_3V6 电源。它从充电器获取电力，并使用 3.6V 输出将其发送到降压升压调节器。3.6V 输出被整个系统中的 LDO 和降压调节器降压，以给其他功能供电。使用集成负载开关和可编程逻辑的 Silego 功率监控器 IC 查看系统按钮和 VBUS 以确定何时为 3.6V 稳压器提供电源以及何时给系统上电。



电源控制器是一个带集成 PFET 负载开关的可编程逻辑设备。电源控制器与 V\_SYS 串联插入，并作为电源开关，能够隔离电池与下游设备。仅在声称了外部信号时，电源控制器才能运行内部时钟。这在负载开关打开时，会限制电池的漏电流，从而限制充电器、电量和电源控制器的漏电流。

配置电源控制器的逻辑，以在检测到 VBUS 或在按下系统按钮 0.5 秒后启用电源。两个输入都被上拉至 V\_SYS。通过强力上拉将 nVBUS\_DET 信号拉升，确保在应用 VBUS 时，通过上拉将浮动的 V\_SYS 轨放电至 0.9V 以下。这保证了如果应用 VBUS 之前不存在任何电池或 USB 电源，电源控制器会进入上电复位状态。

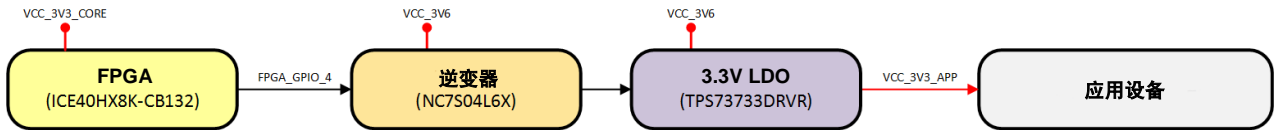
如果不存在 VBUS 且系统按钮按下 4 秒，电源控制器开关将关闭。在未按下系统按钮时，MCU 还可通过将 MCU\_GPIO\_3 调低 0.5 秒，来关闭系统电源。

控制器系统按钮的输入 BTN\_SYS\_CONT 通过 BTN\_OUT 信号传递给 MCU。BTN\_OUT 是一个开漏输出，使用 MCU 上的内部上拉进行拉升。BTN\_SENSE 用于检测连接至 BTN\_OUT 的电线上的电压。如果未按下系统按钮也未声明 BTN\_OUT，则输往 MCU 的电压应被拉高。MCU 可选择将信号电压调低。电源控制器可通过 BTN\_SENSE 检测该状况，并在该状况持续时间达 0.5 秒时断开电源。

一旦内部负载开关闭合，3.6V 稳压器的输入端口就会被施加电源，3V6\_EN 信号在声明前会延迟 15 ms。该 15 ms 的延迟可确保在 3.6V 稳压器尝试接通电流从而为系统其余部分供电之前，V\_SYS 轨已完全上电，且充电器输出电容和降压/升压输入电容均已完全充电。

### 3.3V 应用电源

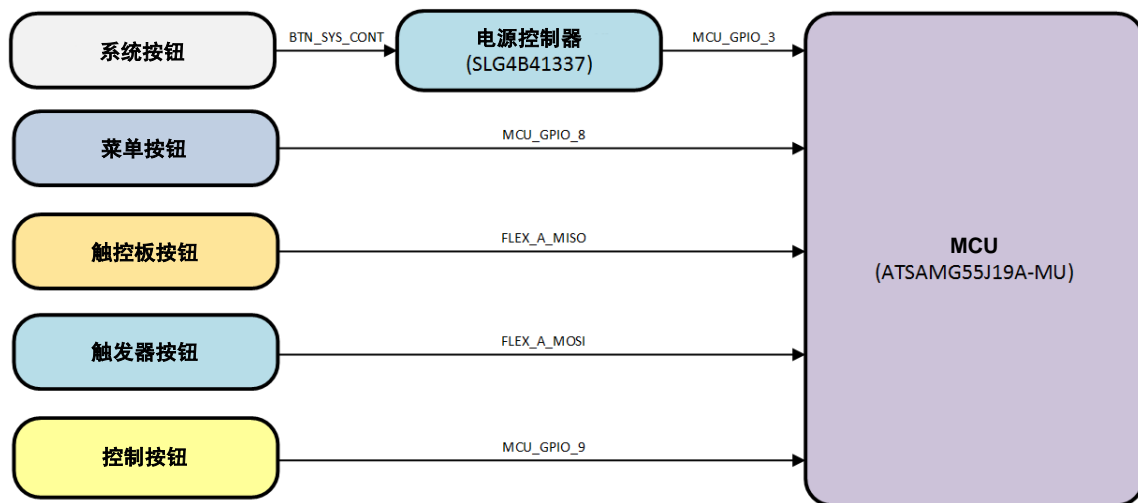
应用电源轨用于将用户控件与系统核心隔离开来，同时，还提供了一种用于关闭与应用有关的外围设备（此类设备可能不具低功耗状态）的途径。



FPGA\_GPIO\_4 信号通过逆变器连接到 3.3V LDO 上的使能输入。该逆变器可保证 LDO 默认为禁用状态，直至配置了 FPGA 且 MCU 将 FPGA\_GPIO\_4 输出设定为低。

### 按钮

电气系统的默认配置支持五个瞬时轻触开关。这些开关分别被分配给 SteamVR™ 系统、菜单、触控板、触发器和控制按钮输入。

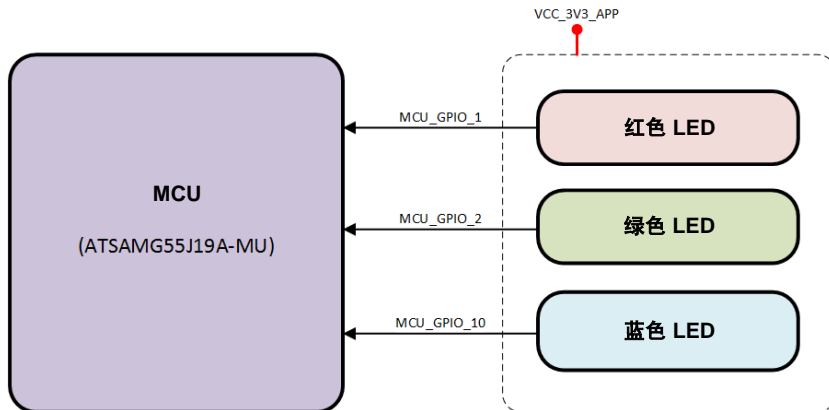


系统按钮通过电源控制器传递。电源控制器将系统按钮用作电源开关。按住此按钮 0.5 秒即可给系统上电，按住此按钮 4 秒即可关闭系统。应用 USB 电源始终会给系统上电。移除 USB 电源后，系统将返回系统按钮所定义的最后一种状态。

MUC 的所有按钮输入都会在内部上拉并去抖。

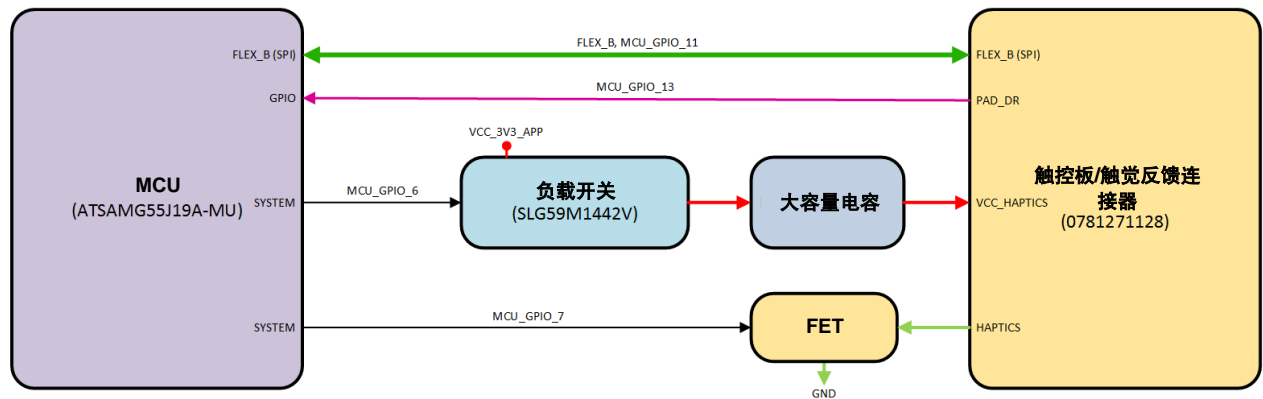
## LED

MCU 上连接有一个用于指示状态的 RGB LED，该 LED 由应用电源轨驱动。



## 触控板和触觉反馈

触控板和触觉反馈组成一个模块，并通过 SPI 和 GPIO 连接到 MCU。触控板外围设备是一个电容式触控设备，使用时将手指放置在触控板的圆形表面上。触觉反馈是一种感应式机电装置。

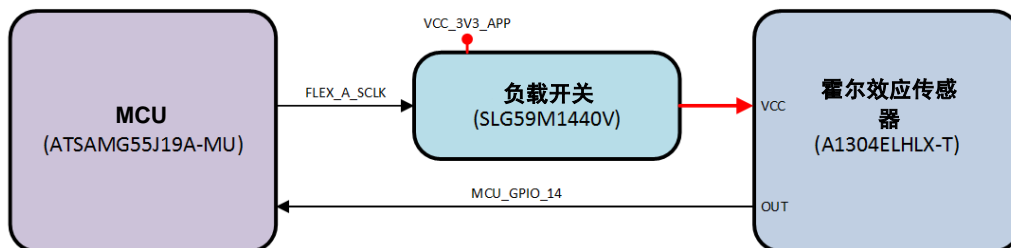


触控板模块通过一个低插入力 FPC 插座连接到 MCU。触控板通过 FLEX\_B SPI 总线与 MCU 通信，并分别使用 MCU\_GPIO\_11 和 MCU\_GPIO\_13 作为发送给 MCU 的芯片选择信号和数据就绪中断信号。

触觉传感器由 VCC\_3V3\_APP 通过负载开关供电，此 VCC\_3V3\_APP 为触觉模块旁边的大容量电容充电。该大容量电容可存储电荷，在低端 FET 被 MCU\_GPIO\_7 激活时快速驱动触觉反馈。为了防止大容量电容充电时系统电源过载，PFET 负载开关将触觉模块的电源轨与内部转换速率限制隔离开来。

## 模拟触发器

如果控制器的触发器包含磁铁，MCU 可通过霍尔效应传感器监控其位置。

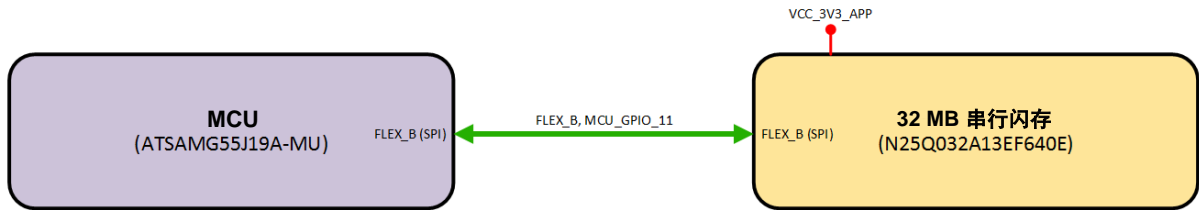


如果仅在对输出进行采样时才驱动霍尔效应传感器，则此传感器的功耗会有所降低。MCU 并不将 FLEX\_A 用作 SPI 总线，而是用作 GPIO。FLEX\_A\_SCLK 是为霍尔效应传感器供电的使能信号，并使用 MCU 的内部 ADC 对输出进行采样。如果 VCC 未降至足够低，那么在重新施加 VCC 之前，霍尔效应传感器会处于闭锁状态。负载开关包括一个输出负载转储电阻以对霍尔效应传感器上的旁通电容器放电，从而保证传感器在下一次负载开关闭合时正确上电。

## HMD 功能

### 串行闪存

HMD 需要外部串行闪存来存储显示屏的校准数据。



串行闪存使用 FLEX\_B SPI 总线连接到 MCU。该总线与控制器应用中的触控板所使用的 SPI 总线为同一总线。

### 其他 HMD 外围设备

HMD 还需要其他外围设备，但在参考设计中未包含此类内容。应根据设备特有的需求来选择这些设备。

## 接口

下面的接口表列出了电气系统中各个设备之间的连接。

光学传感器	
SENSOR_A[32:1]	留待后用。
SENSOR_B[32:1]	传感器包络信号。
SENSOR_VCC[4:1]	四个电源组，每个为八个传感器供电。
SENSOR_VCC_EN[4:1]	由 FPGA 发往至四个传感器电源轨的使能信号。
FPGA GPIO	
FPGA_GPIO[10:5]	可用于应用程序 I/O 扩展和辅助时间戳输入。
FPGA_GPIO[4]	可用于应用 I/O 扩展。通常用于应用电源使能。
FPGA_GPIO[3:1]	可用于应用程序 I/O 扩展和辅助时间戳输入。
无线电 GPIO	
RADIO_GPIO[8:1]	留待后用。
NFC_P, NFC_N	留待后用。
NRF_FPGA_GPIO[2:1]	留待后用。
FLEX_C (SPI)	
FLEX_C_SCLK	无线电和 IMU SPI 接口的串行时钟。
FLEX_C_MOSI	无线电和 IMU SPI 接口的 MCU 数据输出。

FLEX_C_MISO	无线电和 IMU SPI 接口的 MCU 数据输入。
NRF_nCS	无线电片选。
NRF_DR	无线电到 MCU，数据就绪中断信号。
IMU_nCS	IMU 片选。
IMU_INT	IMU 到 MCU，数据就绪中断。
<b>FPGA</b>	
FPGA_SCLK	FPGA SPI 接口的串行时钟。
FPGA_MOSI	FPGA SPI 接口的 MCU 数据输出。
FPGA_MISO	FPGA SPI 接口的 MCU 数据输入。
FPGA_nCS	FPGA 片选。
FPGA_DATA_RDY	FPGA 到 MCU，数据就绪中断。
FPGA_nRESET	FPGA 配置逻辑的重置信号。
FPGA_CDONE	发送给 MCU 的 FPGA 信号，指示配置状态。
FPGA_CRESET	强制 FPGA 进入配置重置。
FPGA_MCU_1	保留。
FPGA_MCU_2	保留。
FPGA_MCU_3	保留。
<b>MCU_GPIO</b>	
MCU_GPIO_1	红色 LED
MCU_GPIO_2	绿色 LED
MCU_GPIO_3	控制器：系统按钮输入。电源按钮。
MCU_GPIO_4	用于调试和制造测试的 UART TX。
MCU_GPIO_5	用于调试和制造测试的 UART RX。
MCU_GPIO_6	控制器：触觉电源使能。
MCU_GPIO_7	控制器：触觉驱动信号。
MCU_GPIO_8	控制器：菜单按钮，ISP 按钮输入。 HMD：ISP 按钮。
MCU_GPIO_9	控制器：握持按钮输入。
MCU_GPIO_10	蓝色 LED
MCU_GPIO_11	控制器：触控板片选。 HMD：闪存片选。
MCU_GPIO_12	未使用
MCU_GPIO_13	控制器：触控板到 MCU，数据就绪中断。
MCU_GPIO_14	控制器：模拟触发器 ADC 输入。
FLEX_A_MISO	控制器：配置为 GPIO，触控板按钮输入。
FLEX_A_MOSI	控制器：配置为 GPIO，触发器按钮输入。
FLEX_A_SCLK	控制器：配置为 GPIO，模拟触发器使能输出。
<b>FLEX_B</b>	

FLEX_B_SCLK	控制器：触控板 SPI 串行时钟。 HMD：闪存 SPI 串行时钟。
FLEX_B_MOSI	控制器：触控板数据输入。 HMD：闪存数据输入。
FLEX_B_MISO	控制器：触控板数据输出。 HMD：闪存数据输出。
<b>I2C_BATT</b>	
SCL	MCU、充电器和电量计之间的 I2C 时钟。
SDA	MCU、充电器和电量计之间的 I2C 数据。
BATCHG_INT	充电器和电量计到 MCU，数据就绪中断。
<b>USB</b>	
USB_D_P	USB 正数据信号。USB 全速。
USB_D_N	USB 负数据信号。USB 全速。
USB_VBUS	USB 5V 电源。
<b>MCU_DEBUG</b>	
SWDIO	MCU 单线数据。
SWCLK	MCU 单线时钟。
nRST	MCU 重置。
SWO	MCU 单线输出。
<b>RADIO_DEBUG</b>	
SWDIO	无线电单线数据。
SWCLK	无线电单线时钟。
nRST	无线电重置。
SWO	无线电单线输出。