GPIB 总线 Device 实现的总结报告
刘志勇, https://liuzhiyong.cn/
2025/1/18

实现 GPIB 总线的 Device，硬件为：
GPIB 的引脚 10K 上拉电阻到 3.3V，并且连接到 STM32F446 单片机。

引脚配置：
- 输入：有上拉电阻。
- 输出：OD（开漏），有上拉电阻。

引脚功能：
- ATN：起效（Assert）的时候，获取命令字节（command）。没有起效的时候，为数据消息的字节。
- NRFD（没有准备好接收）、DAV（数据放置在总线 DIOx 了）、NDAC（没有接收数据）、EOI（标记结束的字节）：通信为——准备好接收了，字节放置在总线了，接收了。然后最后一个字节标记 EOI 起效。
- DIOx：一个字节的八个比特。
- 其他引脚：没用。

操作流程：
- 检测到 ATN 起效（Assert），读取命令字节（command），我们关心的只有五个命令：Untalk, Unlisten, My Talker Address x, My Listener Address y, Secondary Address z。含义分别为：停止 talk（释放总线），停止 listen（释放总线），x 开始 talk，y 开始 listen，子地址 z。
- ATN 失去起效（Assert），
  - 如果是命令我的设备 Listen，那么一定要立刻开始读取总线的数据。如果不立刻开始读取，会丢失的一个字节。
  - 如果是命令我的设备 Talk，发送的最后一个字节，要标记 EOI，而且 EOI 起效（Assert）时间大于（包含）DAV，否则 Keysight/NI 上位机显示超时。
  - 如果命令为子地址，那么要立刻释放总线（DIDS），否则 NI 会扫描出一堆错误的子地址。

通常顺序：
- 上位机 Controller 要所有设备释放总线，然后要 Controller 来 Talk，要我的 Device 来 Listen：然后发送命令，例如*IDN?
- 上位机 Controller 要所有设备释放总线，然后要 Device Talk - Controller Listen：然后接收数据，例如设备名称（IDN 的回复）

参考代码：
AR488 最主要是 attnRequired, readByte, writeByte 三个函数。它有很多其他功能，对我们是没用的。
改进的地方：
- 发现命令为子地址（Secondary Address），都释放总线（DIDS），因为我们不用。
- attnRequired 检测到我们的 Device 要 Listen，那么标记 IsReadByteAfterATN 为 true，在 readByte 里面检测到 ATN 失去起效（Asserted），立刻开工循环读取（如果延迟，就丢失第一个字节）。
- attnRequired 检测到我们的 Device 要 Talk，那么发送数据，发送的最后一个字节，要标记 EOI，而且 EOI 起效（Assert）时间大于（包含）DAV，否则 Keysight/NI 上位机显示超时。
  - 设置 DIOx 之后，DAV 起效（Assert）之前，最好延时一段时间例如 2us。“The T1 delay lets the data lines settle to stable values before they are read.”