Understanding I3C Communication Protocol & Haptic Feedback Driver Design

Introduction: Haptic Feedback and I3C

- **What is I3C?**
  - The **BOS1921** piezo haptic driver IC provides versatile haptic feedback, commonly used in devices requiring vibration patterns to interact with users.
  - While the development board is costly (~240€), the main IC, BOS1921, is more affordable (~4€), making it attractive for project development.
  - This project involves creating a **breakout board** for the BOS1921 and investigating **I3C**, which is a successor to the I2C communication protocol.

Breakout Board Creation

- **Designing the Breakout Board**
  - **Step 1**: Start by reviewing the **datasheet** for the IC to gather the schematic, PCB design, and component lists.
  - **Step 2**: Convert the schematic to a **PCB design**. Using a **4-layer PCB design** for optimal component placement—capacitors and coils are placed as close to the IC as possible.
  - **Step 3**: Order PCBs and begin assembly. Use **solder paste** and a **mini hot plate** for reflow soldering, adding **male headers** and **terminals** for testing.
- **Troubleshooting Issues**
  - After assembly, **no boards worked** initially due to excessive current draw.
  - **Thermal camera** helped identify the issue: a wrongly placed capacitor.
  - The capacitor was flipped, causing the issue, but after flipping it, the boards powered up successfully.
- **Final Test**
  - The boards were tested by connecting them to the development board, playing back waveforms using the onboard software. The **haptic actuator** performed correctly, confirming the breakout board's functionality.

I2C vs. I3C Communication Protocol

- **I2C Overview**
  - **I2C** allows communication with a maximum frequency of **1 MHz**, a widely used communication protocol in electronics.
  - It uses **open collector drivers** with pull-up resistors for signal transitions.
- **I3C Overview**
  - **I3C** allows a **maximum frequency of 12.5 MHz**, offering a much higher data transfer rate than I2C.
  - Key differences include the use of **push-pull drivers** (instead of open collector) for more efficient signal transitions and lower power consumption.
- **Main Differences Between I2C and I3C**
  - **Timing**: I3C offers faster data transfer (12.5 MHz) compared to I2C (1 MHz).
  - **Dynamic Addressing**: I3C assigns dynamic addresses during initialization, preventing address collisions that can occur with I2C.
  - **Common Command Codes (CCC)**: Allows for fast commands like resetting addresses or setting new ones.
- **Advanced I3C Features**
  - **Hot Join**: I3C devices can be added or removed without resetting the microcontroller, a feature not supported by I2C.
  - **In-Band Interrupts**: Eliminates the need for an external interrupt pin, allowing interrupts to be communicated through the data stream.

Conclusion and Verdict

- **I3C Advantages**
  - **Higher Data Transfer Rates**: I3C provides faster communication (12.5 MHz vs. 1 MHz).
  - **Dynamic Addressing**: Reduces the risk of address conflicts and simplifies integration.

- **Hot Join & In-Band Interrupts**: Improve functionality and reduce wiring complexity.
- **Challenges with I3C**
    - **Complexity**: I3C is more complicated and requires special peripherals to implement, which are not widely available in microcontrollers yet.
    - **Current Microcontroller Support**: Only a few microcontrollers currently support I3C.
- **I2C Sufficiency**
    - I2C remains **sufficient for many projects**, including the BOS1921 haptic driver, which uses I2C for communication on its development board.
- **Future of I3C**: While I3C has potential, it is not yet as widespread as I2C. For now, **I2C** will likely remain the go-to protocol for many applications.

Takeaways

- **I3C** is a powerful successor to **I2C**, offering faster speeds, better addressing, and fewer wires for complex sensor systems.
- The **BOS1921** can work with both I2C and I3C, but for most projects, **I2C** will still be the simpler and sufficient choice.
- **Breakout board** creation and troubleshooting are essential skills in project development, helping to create affordable, customizable solutions.