An LED-to-LED Visible Light Communication System with Software-Based Synchronization

Stefan Schmid\textsuperscript{1,2}, Giorgio Corbellini\textsuperscript{2}, Stefan Mangold\textsuperscript{2}, Thomas R. Gross\textsuperscript{1}

\textsuperscript{1}ETH Zurich, \textsuperscript{2}Disney Research Zurich
Motivation – Networking Toys

- **Requirements:**
  - Low bit rate communication
  - Short distances
  - Cost-effective

- **LED-to-LED communication:**
  - Fulfills requirements
  - Communication is visible
  - No radio waves
  - Reuses already existing components
  - Low-complexity
Outline

• Motivation

• Technology
  • LED as a Transmitter
  • LED as a Receiver
  • Synchronization

• Implementation and Platform

• Evaluation

• Conclusions
Transmitter

• On-off keying
  - LED on → symbol ONE
  - LED off → symbol ZERO

• Requirements:
  - No flickering
  - Constant brightness

• Example: Encode 1 bit directly with a symbol
  - 1-byte message: \textbf{1000 0001} → ON OFF OFF OFF OFF OFF OFF OFF ON
  - Long OFF-sequence changes the perceived brightness
  - Even longer OFF-sequence introduces flickering
Manchester Coding

- 0 → 1 0 → ONE ZERO
- 1 → 0 1 → ZERO ONE
- Uniform distribution of ONE and ZERO symbols
- Redundancy
LEDs as Light Receivers

• LEDs can be used as light receivers [Dietz et al, 2003]

• LED used as a receiver:
  • Reduces complexity
  • Cost-effective

• Limited light sensitivity

But
  • Short range communication
  • Low data rates

still possible

Receiver

• LED is used in reverse bias → capacitor
• LED (cathode) is periodically charged
• Measure residual voltage after a certain time period
Threshold

• Receiver needs a threshold to decode the residual voltage level to a ZERO or ONE

• Requirements:
  • Largest distance from both ZERO and ONE symbol
  • Adaptive to ambient light

• Mean of recent measurements:
  • Contains ZEROs (ambient light) and ONEs (LED generated light)
  • ZEROs and ONEs are uniformly distributed (Manchester)
  • Provides a good threshold if constantly updated
Synchronization (1)

- **Goal**: Match the start of a transmitted symbol with the start of a measurement period
- **Synchronization pattern**: Pairs of ONE and ZERO symbols
- **Shift until measured light for ONE symbols is equal**
Synchronization (2)

- Transmitter
- Receiver
- ONE detected
- Start of measurement
- Offset
- Synchronize

Diagram with timestamps and measurement details.
Always «on»

• **Problem:**
  • LED appears as «off» in reception mode
  • Not possible to measure the photo-current while the LED is on

• **Solution:**
  • LED multiplexing \(\rightarrow\) alternating measuring and light emitting slots
  • No change in transmitter, thanks to Manchester coding (redundancy)
  • LED perceived as «on» \(\rightarrow\) 1 kHz signal

\[
\text{\begin{array}{cccccccccc}
\text{e} & \text{m} & \text{m} & \text{e} & \text{e} & \text{m} & \text{m} & \text{m} & \text{e} \\
\end{array}}
\]
\[
e = \text{light emitting slot} \quad m = \text{measuring slot} \quad 500\mu\text{s}
\]
Challenges (1)

- Receiver only measures light slots:
  - Ambient light never measured
  - Computed threshold is equal to the symbol ONE
  - Decoder cannot detect all ONEs anymore

- Halving the computed threshold provides a good approximation (ambient light value 0)
Challenges (2)

Broken synchronization:

- Two consecutive measurements are **different** for the **same amount** of received light
- Different preconditions for measuring circuit (ADC)
- Measurement deviation depends on LED

**Correction:**

- Deviation is a linear function
- Correction computed from preceding measurement
Outline

• Motivation

• Technology
  • LED as a Transmitter
  • LED as a Receiver
  • Synchronization

• Implementation and Platform

• Evaluation

• Conclusions
Hardware Platform

• Various Arduino boards for prototyping

• ATmega 328P microcontroller
  • 8 bit
  • 32K program storage
  • 2K RAM
  • 8 ADC pins (multiplexed)

• In-house PCB solution
  • Based on Arduino layout
  • Smaller form factor for toy integration
Implementation

• Flexible software implementation (C++, AVR-g++) of physical and MAC layer

• Hardware:
  • ADC
  • Timers

• Software:
  • Synchronization
  • Encoding / Decoding
  • MAC

• MAC
  • Acknowledgments/ Retransmissions
  • CRC
Evaluation

• Robustness for typical toy scenarios

• Data throughput for:
  • Different (short) distances
  • Different data rates

• Testbed setup:
Evaluation Results
Conclusions

• Short-range communication system for toys
• VLC based on only LEDs
  • Low-cost
  • Low-complexity
• Software driven physical/MAC layer
• Short distance communication
• 100-200 b/s data rates
Thank you!