Abstract

- Use low-power LEDs to transmit data optically via freespace (air); the signals will be captured by a camera-equipped receiver.
- Design a transmission protocol to optimize this type of optical communication, both in terms of speed and reliability. We will demonstrate, with this design, a feasible communication solution that will be useful for mote-based systems.

Technical Approach: camera

- The physical operation of the camera is based on 3 different color filters and light-intensity sensors. Each pixel is physically a single light-intensity sensor shielded by a red, green, or blue filter. Each sensor, therefore, measures the intensity of its corresponding filter’s light at its exact location.
- The intensity of the indicated color at each pixel is digitized and encoded in 8 bits for Y8 encoding, the default, or 16 bits for Y16 encoding. Therefore, for the Y8-RGGB color encoding of a 640x480 image, the raw data requires 640x480x8 bits, or 300KB.
- The code behind the camera had three main tasks:
  1. Determine where the transmitters are located. Not done in real time because involves analyzing several images with 300,000 pixels in each image.
  2. Set a threshold for on and off. Within a given range of the maximum intensity, the light is considered on. Otherwise, if outside of this range, it is considered off.
  3. Real-time analysis begins. When light is seen to be on, a 1 is communicated to the higher level. Otherwise, a 0 is.

Technical Approach: blinker

- LED communicates the message, controlled by a microprocessor.
- Synchronized with camera, which runs at 30 frames per second, so only tries to communicate 30 bits each second.
- The transmission unit had 20KB of external RAM running at 2.0 MHz. Coding was done in C and loaded into memory via a serial connection.
- While perhaps slower than a system using dedicated hardware, added speed was not necessary to operate at 30 bits/sec, the maximum allowed by the receiving node.
- Physical data transmission occurs regularly at a rate of 30 Hz. At each period, the next bit is read off of the queue and output to a parallel output port, with a high voltage representing 1 and a low representing 0.

Protocol

- Now that bits can be transmitted, need way of associating metadata with data - frames.
- Camera and blinker never completely synchronized, so potential for dropped bits. Camera uses a CRC-8 checksum to verify a frame.
- Starting flag of 0x7E (8 bits) designates the start of a frame.
- Need way to make sure byte within frame is not the starting flag. So, every 4 bits are translated to a planned 5 bit value that guarantees there will never be more than 3 1's in a row.
- Frames are cellular, i.e. they have a set width, so the camera knows to only expect 5 bytes after a starting flag has been.

Frame Structure

<table>
<thead>
<tr>
<th>Sentinel</th>
<th>Transmitter ID</th>
<th>S</th>
<th>Seq Number</th>
<th>Data</th>
<th>CRC-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x7E</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmitter ID: 0-255 (8 bit)</td>
<td>S: Sequence number set (1 bit)</td>
<td>Seq number: 0-127 (7 bits)</td>
<td>Data data being sent (8 bits), or max seq number if frame has seq #0</td>
<td>CRC-8 (8 bits) - checksum applied to the previous three bytes. Used by receiver to verify the frame</td>
<td>Bytes in frame mapped special way to avoid putting sentinel in body.</td>
</tr>
</tbody>
</table>

Efficiency

- Low power readings even when LED transmitting. Camera uses much more power, but at least in this setup not as important.
- Data efficiency not as good in case of one blinker - 8-19% depending on the size of the message. Can be improved with multiple blinkers.

Conclusions

- Has the potential of beating 802.11 adapters’ efficiency in terms of power on the transmission end.
- The data efficiency is low, and the 30 bit/s data rate does not help the overhead. However, given coordination between multiple LEDs to broadcast to one receiver, this situation can be improved drastically.