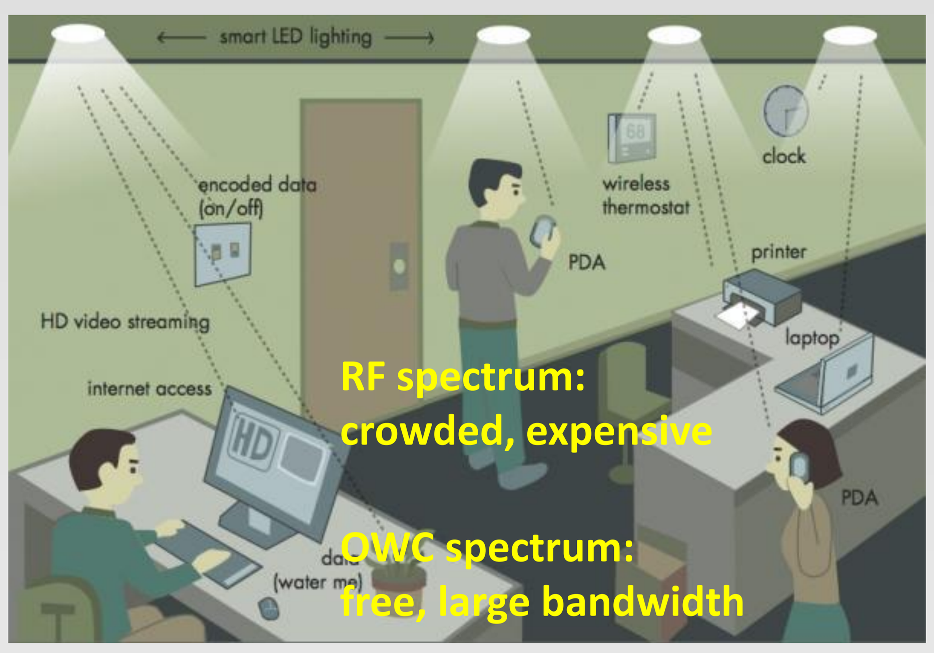
A Software-Defined Radio Implementation of a VLC (Visible Light Communication) System

Seokseong Jeon[1], Young-Joo Suh[1], and Chansu Yu[2]
[1]POSTECH, Korea
[2]Department of Electrical Engineering and Computer Science
Washkewicz College of Engineering, Cleveland State University

Introduction

Visible light communication (VLC) is an emerging field and has lots of implications in terms of **IoT (Internet of Things)** applications. In a VLC system, data is carried by visible light. Therefore, VLC can exploit indoor lighting for data communication and can relieve RF (e.g. Wi-Fi) channels of saturation.



(Source: Boston University)

This work presents an implementation of a VLC transmitter and a receiver based on IEEE 802.15.7 specifications with a **SDR (Software-Defined Radio)** platform based on GNU Radio and Universal Software Radio Peripherals (USRPs). Note that IEEE 802.15.7 is the standard and specifications of optical wireless communication (OWC), which is also known as free space optical communication (FSO) and VLC.

Specifications

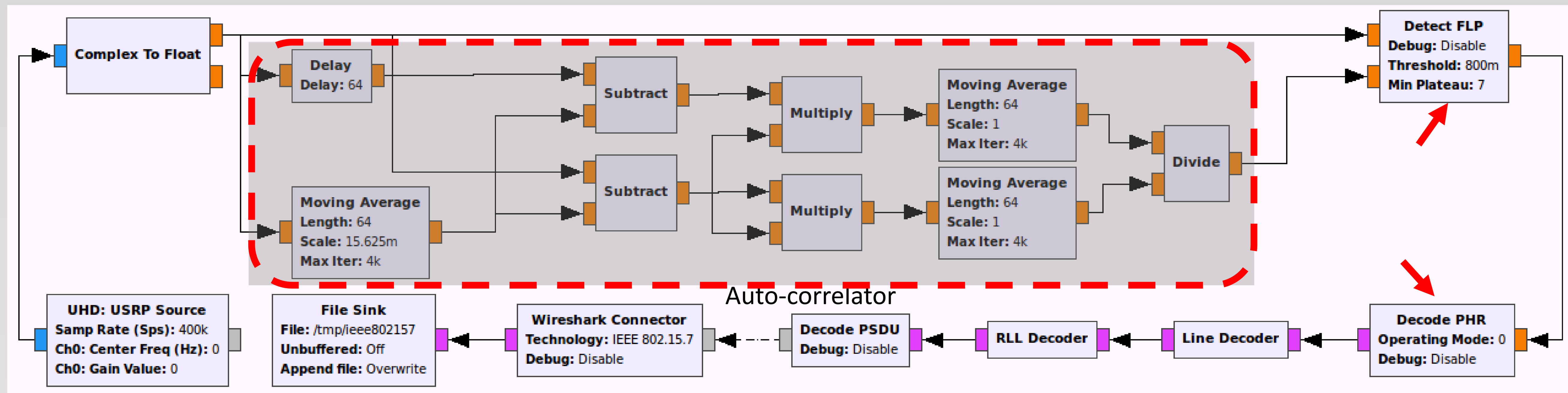
IEEE 802.15.7 has 3 groups of operating modes (**modulation and coding scheme; MCS**) - PHY I, II and III. Due to limitations of USRP front ends, this work focuses on PHY I and II. Each operating mode has a different data rate as described in a table below:

Baseband modulation	Line code	Clock rate	FEC		Data rate (kbps)
			RS (n, k)	CC	
On-off keying (OOK)	Manchester	200 kHz	(15, 7)	1/4	11.67
			(15, 11)	1/3	24.44
			(15, 11)	2/3	48.89
			(15, 11)	-	73.3
			-	-	100
Variable pulse-position modulation	4B6B	400 kHz	(15, 2)	-	35.56
			(15, 4)	-	71.11
			(15, 7)	-	124.4
			-	-	266.6

Operating modes in PHY I group

Implementation

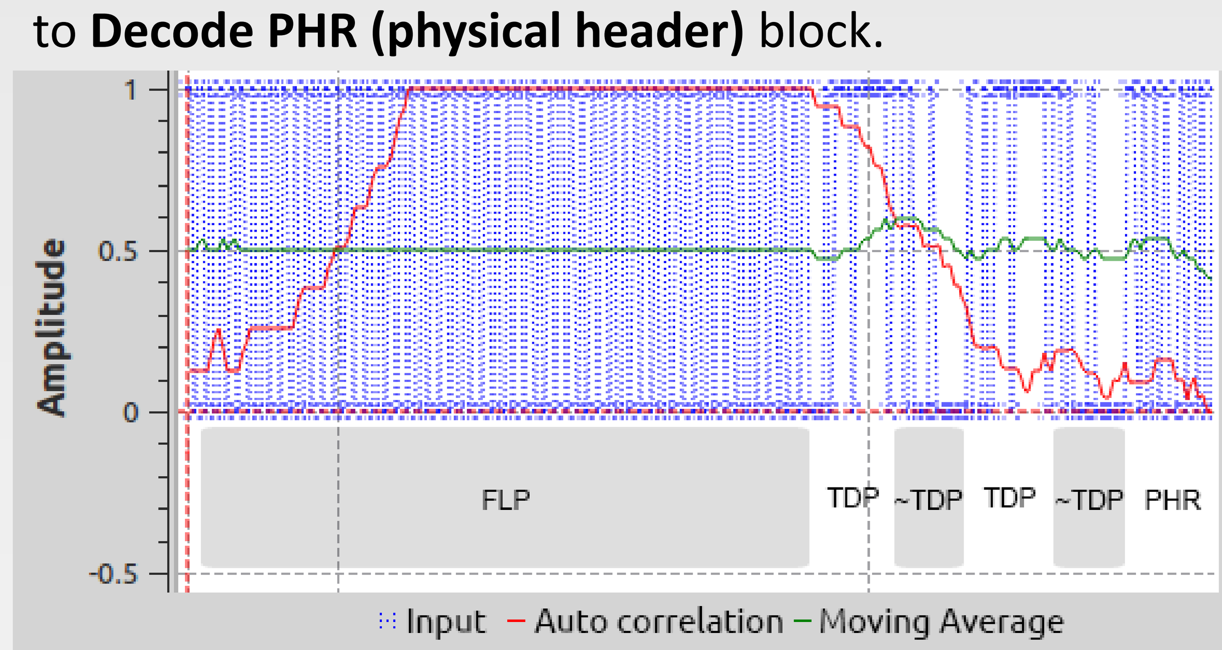
The implementation has been done with GNU Radio and USRPs. In addition, we have built an **LED driver and a photo detecting circuit** as front ends instead of RF antennas. Since detecting and decoding a frame on a receiver side is more complex than generating a frame on a transmitter side, we explains an implementation of the receiver in detail. An overall GNU Radio flow graph of a **"receiver"** is drawn next.



GNU Radio flow graph of a IEEE 802.15.7 receiver

In the GNU radio flow graph shown above, the shaded part implements **fast locking pattern (FLP)** in IEEE 802.15.7, which performs the autocorrelation operation of a received signal to detect a preamble. A high autocorrelation value denotes the presence of a legitimate signal.

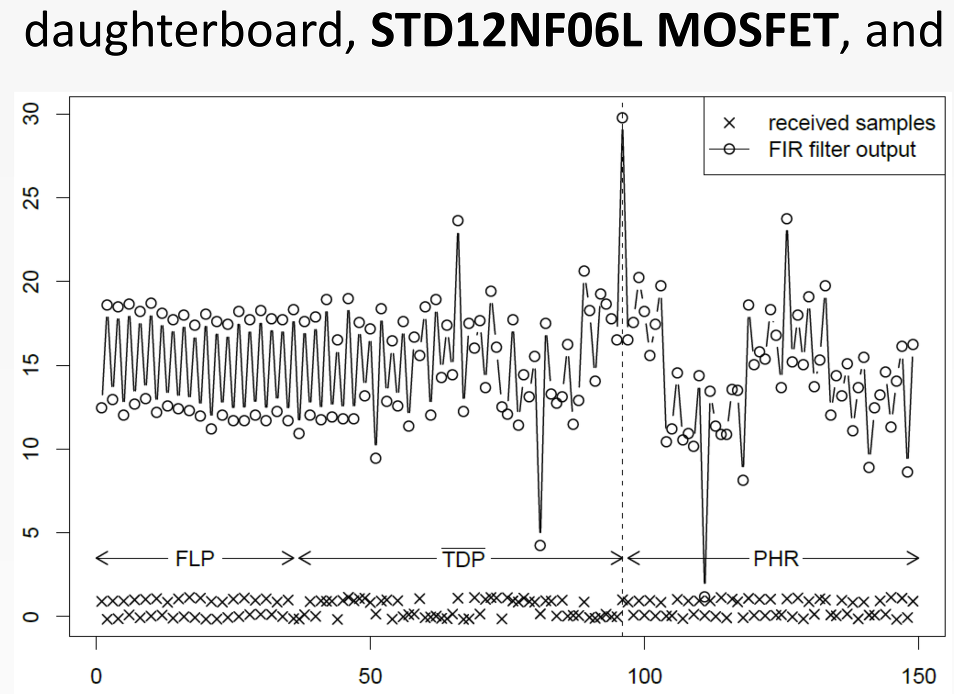
Detect FLP block identifies the duration (plateau) where the autocorrelation value is greater than a threshold. When this duration is longer than the minimum plateau value, it recognizes that there is a frame to be received (see below). The signal is passed to **Decode PHR (physical header)** block.



Autocorrelation of preamble (FLP) shows a long plateau region

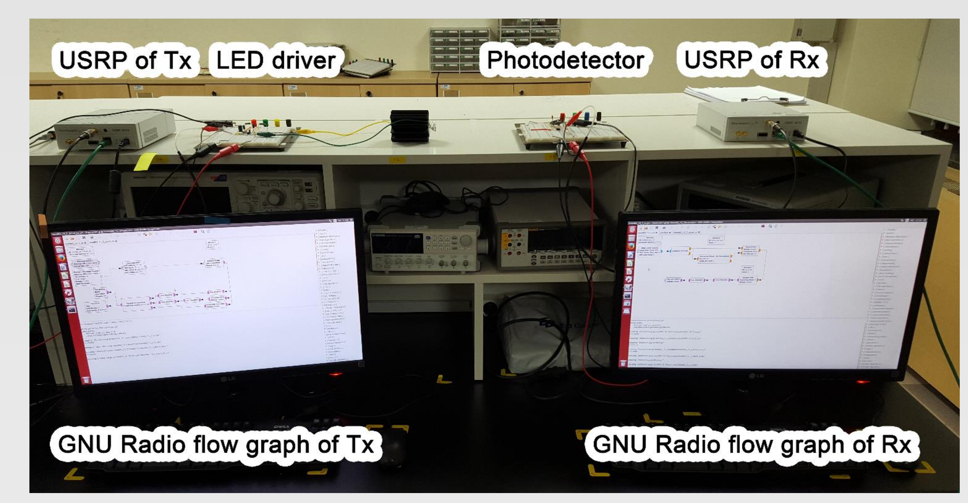
Decode PHR block looks for a beginning of a frame by locating a start frame delimiter, called **topology dependent pattern (TDP)**. A finite impulse response (FIR) filter is used to perform cross-correlation and detect the beginning of a frame (see below). Cross-correlation reaches a peak value when the exact sequence of TDP is received so that the receiver can synchronize and detect the beginning of a frame. After detecting a frame, the receiver can demodulate and decode it with a simple byte array manipulation.

Finally, we have built analog front ends using **LFRX** daughterboard, **Hamamatsu Si PIN diode S6036** (photodiode), **TI OPA134** (Op Amp), **Minicircuits ZFL-500** (coaxial amp) replacing RF antennas for IEEE 802.15.7. For a transmitter, we used **LFTX** daughterboard, **STD12NF06L MOSFET**, and **P4 LED**.



Received samples and its cross-correlation with a peak at x=95

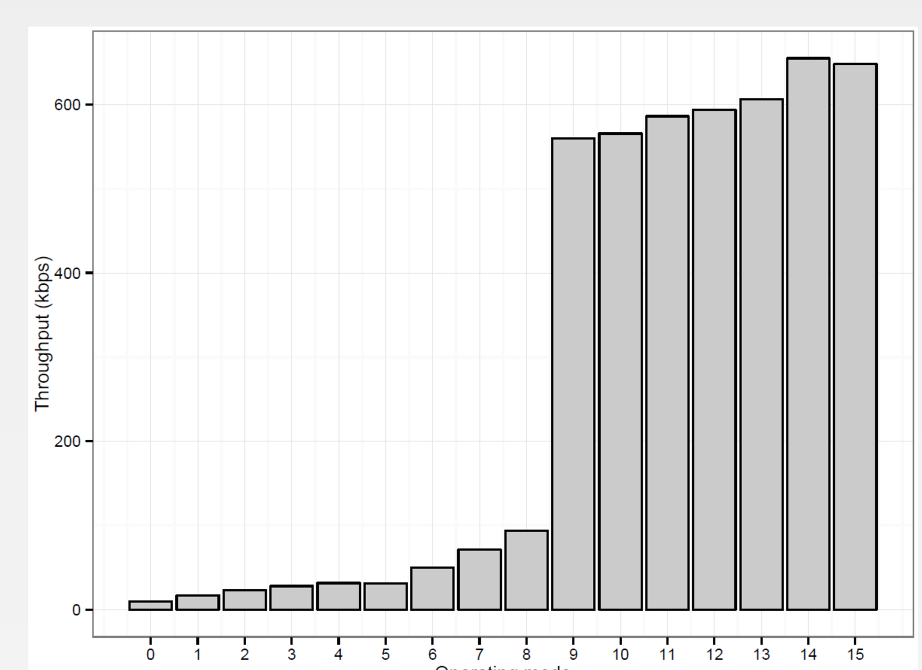
Performance Evaluation



An overall system configuration

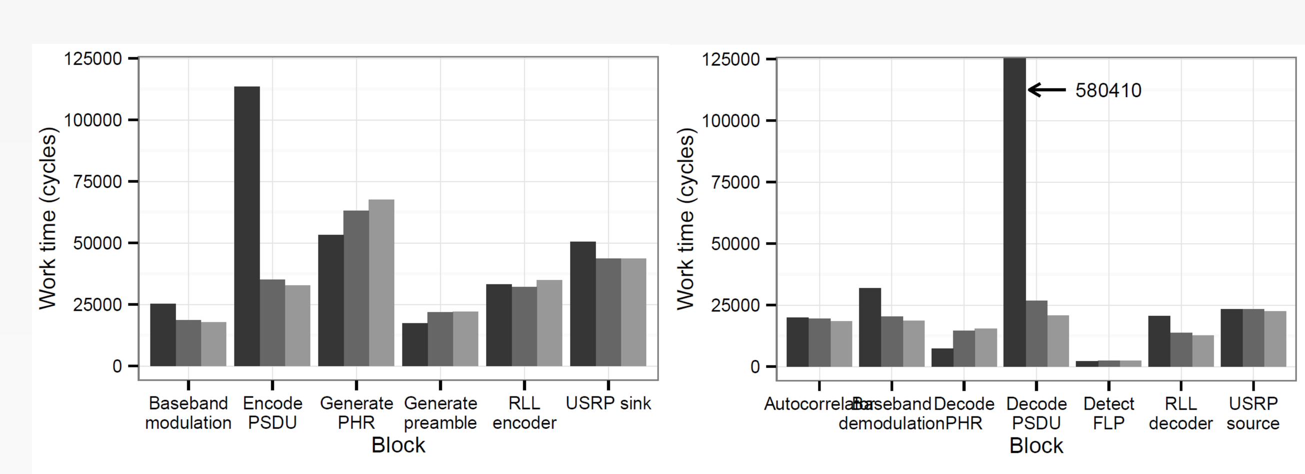
The overall system is shown in the above.

Throughput of each operating mode in PHY I group and a part of PHY II has been evaluated in this study. The result is shown in the figure below:



PHY I throughput of the GNU Radio-based IEEE 802.15.7 VLC system

SDR is flexible but generally takes more computational workload for processing radio or optical signals. To optimize the design, computation time of the transmitter and the receiver has been measured. It was observed that convolutional code computation consumes the most CPU cycles (see below). One of future work is to optimize the convolutional code processing of IEEE 802.15.7.



Work time of each block of the transmitter and the receiver. 25,000 clock cycles correspond to 6.4 us