FPV Streaming Server with ffmpeg

Chan Shik Lim
chanshik@gmail.com

Introduction

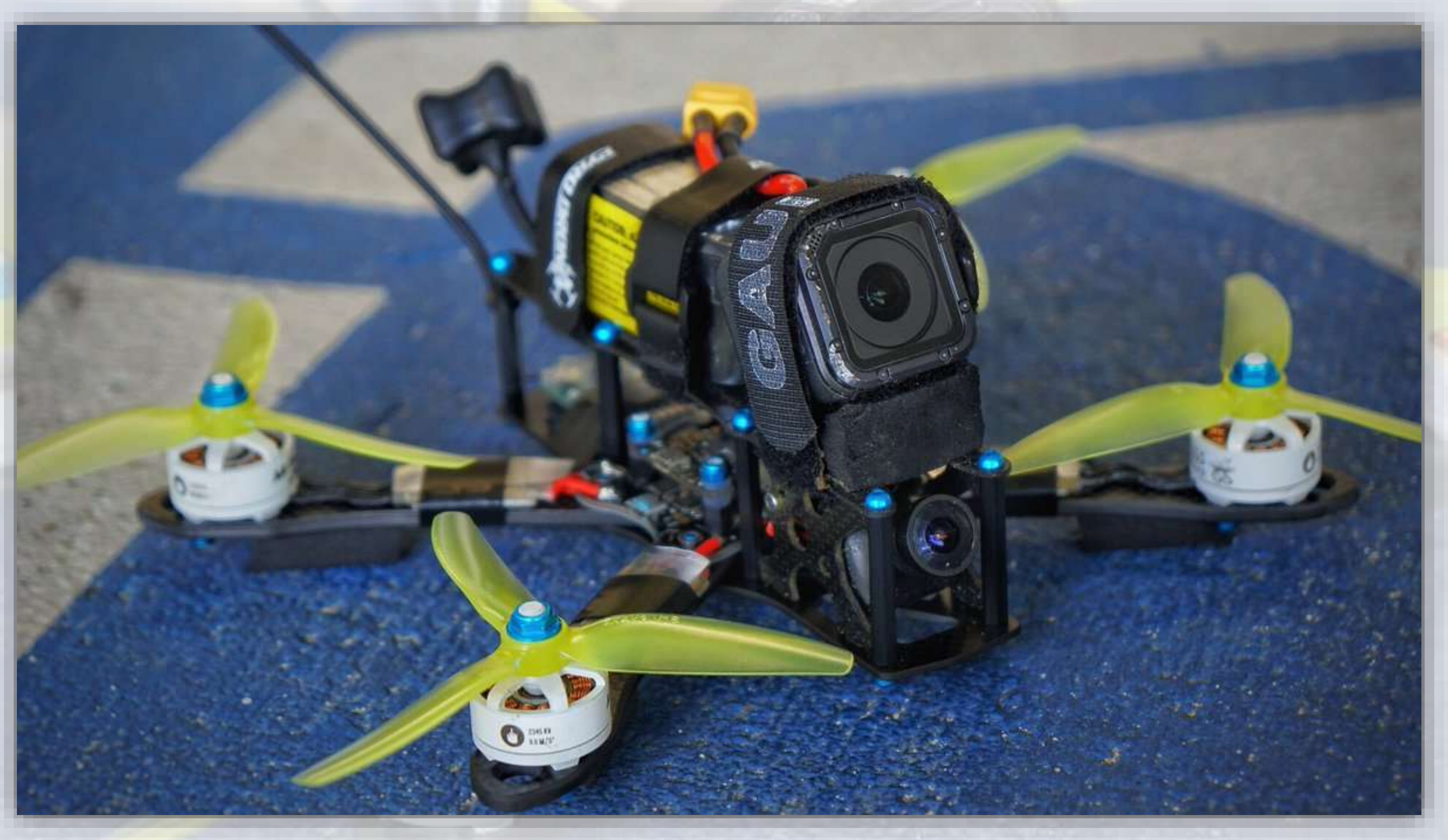
- Chan Shik Lim (chanshik@gmail.com)
- Programmer @ NexCloud
- Go, Python
- System Programming
- Linux, Kubernetes, Prometheus
- **Hiring Programmer for Kubernetes Operator & Prometheus & Monitoring System**

Agenda

- FPV
    - FPV System
    - FPV Channels and Frequency
- FPV Streaming System
    - Build ffmpeg on Windows
    - Incoming Stream Handler
    - WebSocket Handler
    - Client Handler
    - JSMPEG Video Canvas
- DEMO

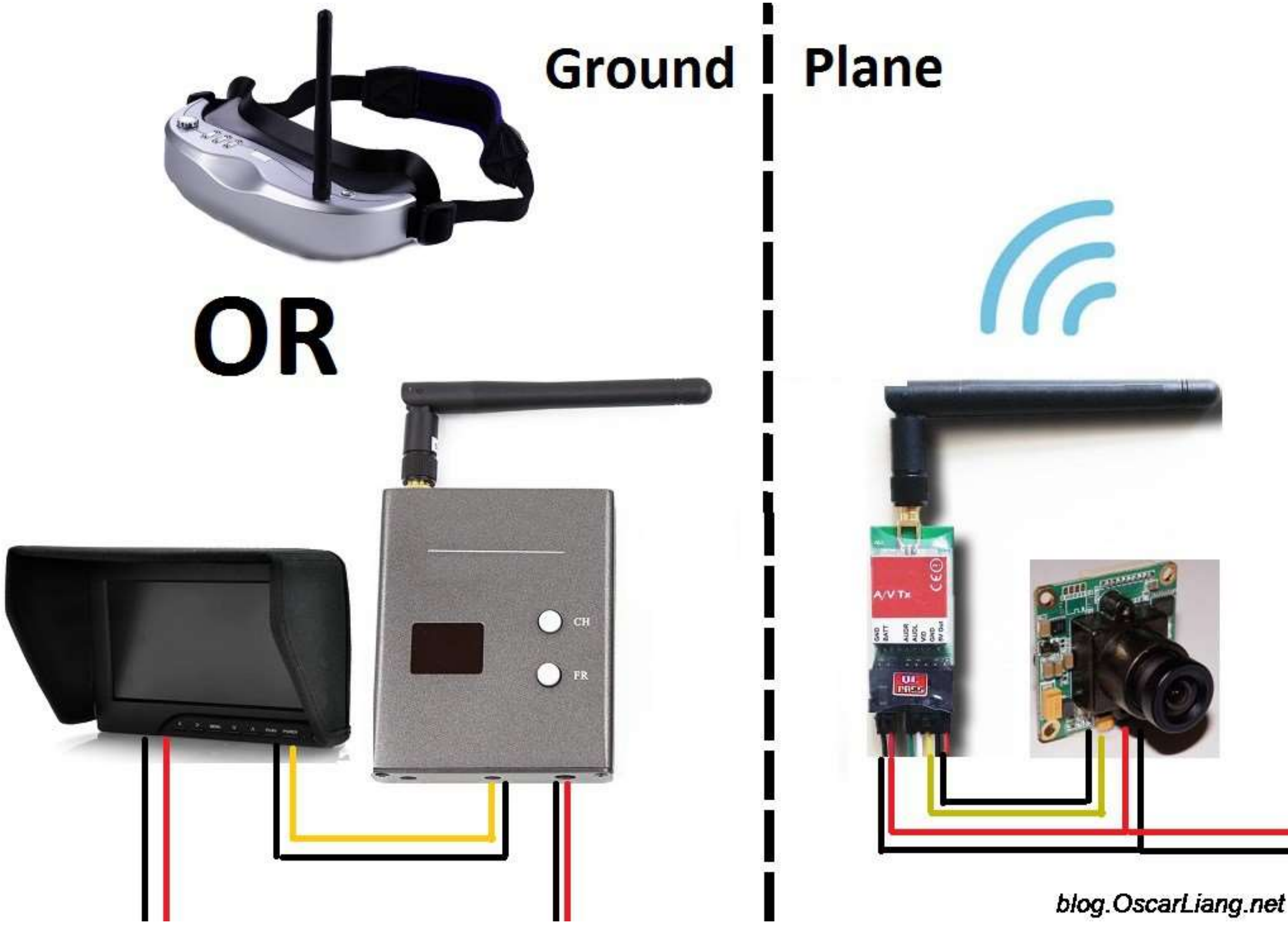
FPV (First Person View)

FPV Drone with ActionCam



FPV System

- **FPV Camera**
- **Video Transmitter** (VTX)
- **Video Receiver** (VRX)
- **FPV Goggles** or **Monitor**
- **Antennas**

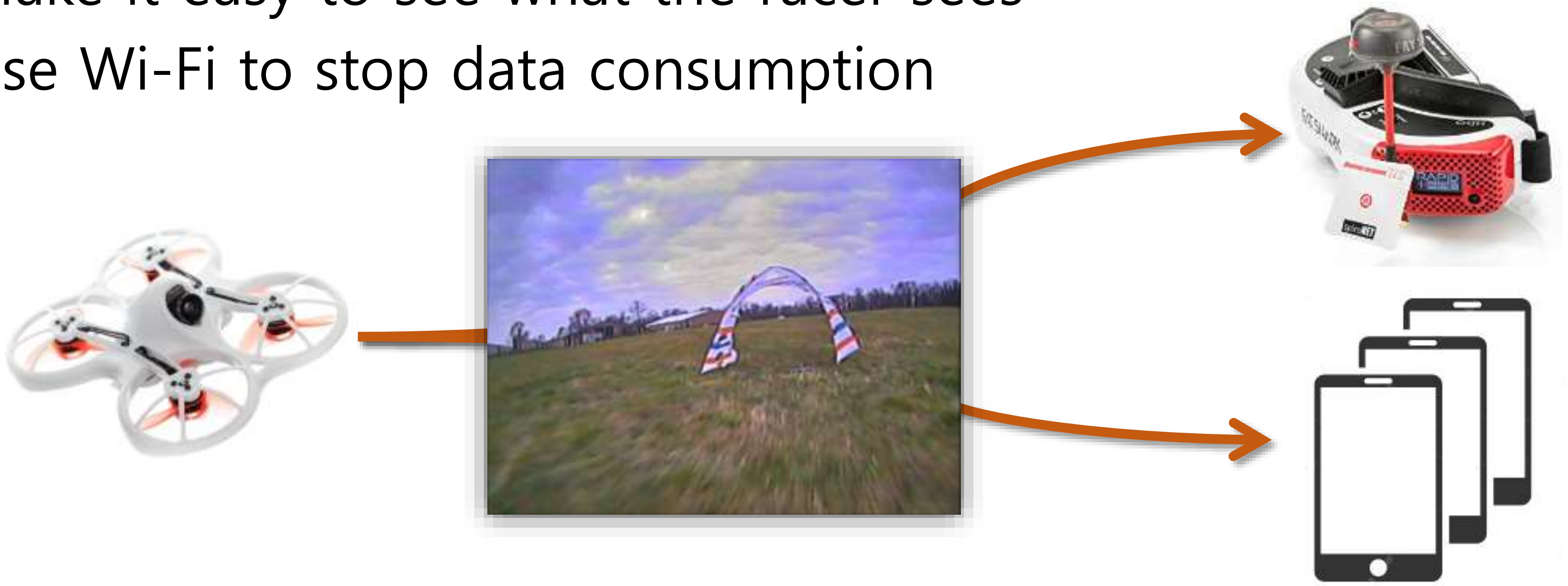


5.8Ghz Channels and Frequency for FPV

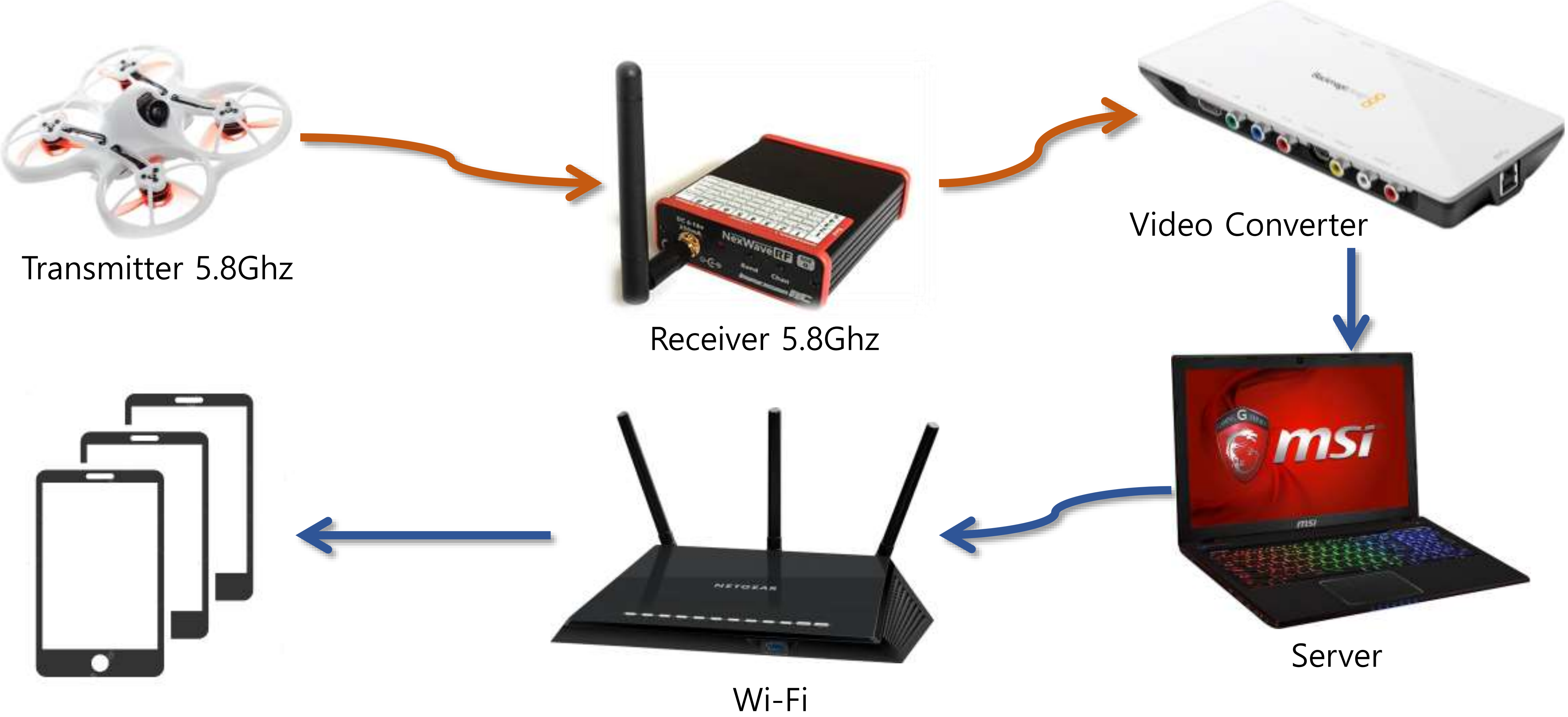
Band	CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7	CH 8
A	5865	5845	5825	5805	5785	5765	5745	5725
B	5733	5752	5771	5790	5809	5828	5847	5866
E	5705	5685	5665	5645	5885	5905	5925	5945
F	5740	5760	5780	5800	5820	5840	5860	5880
C (Race)	5658	5695	5732	5769	5806	5843	5880	5917
D	5362	5399	5436	5473	5510	5547	5584	5621
U	5325	5348	5366	5384	5402	5420	5438	5456
O	5474	5492	5510	5528	5546	5564	5582	5600
L	5333	5373	5413	5453	5493	5533	5573	5613
H	5653	5693	5733	5773	5813	5853	5893	5933

FPV Streaming System (1)

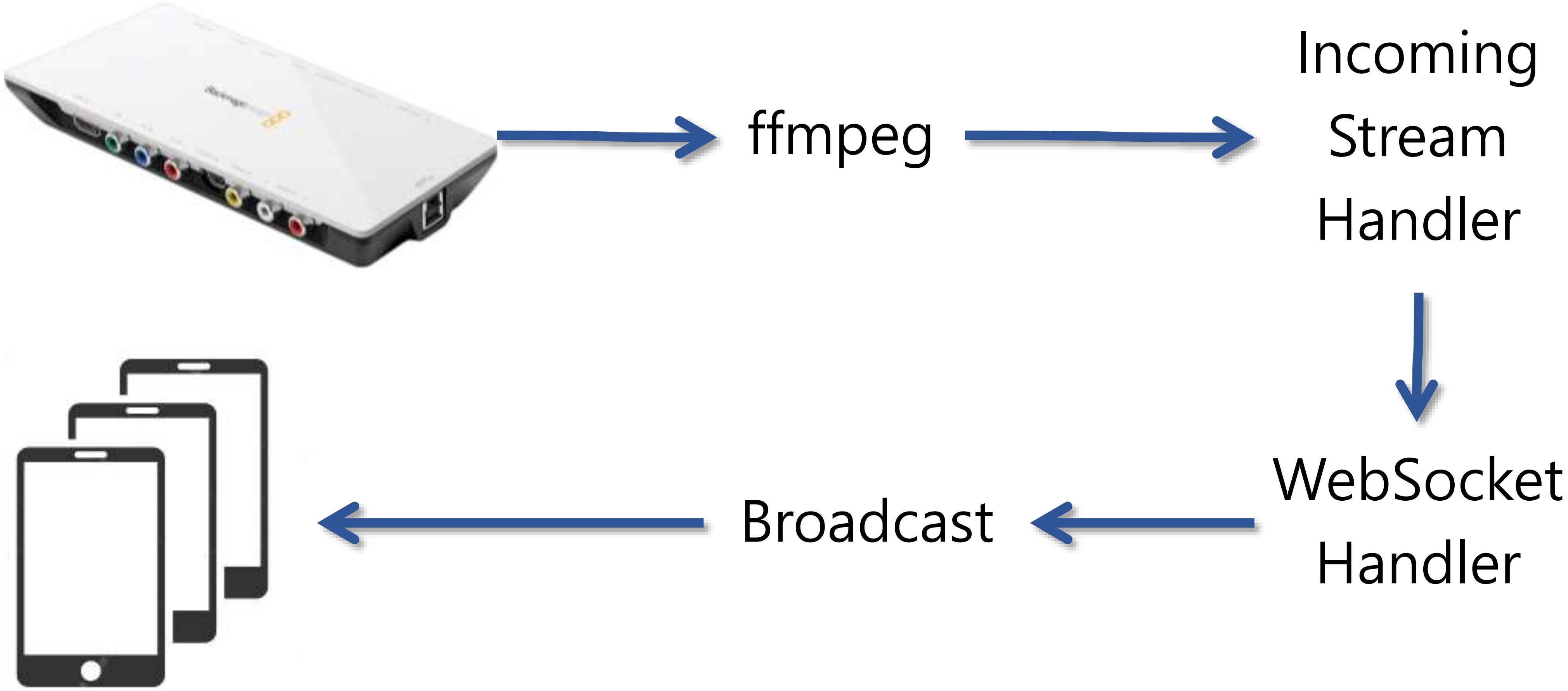
- Share video transmitted in the 5.8Ghz band
- Make it easy to see what the racer sees
- Use Wi-Fi to stop data consumption



FPV Streaming System (2)



FPV Streaming System (3)



ffmpeg

Incoming Stream Handler

WebSocket Handler

Broadcast

Build ffmpeg on Windows

- Enable the Windows Subsystem for Linux optional component
- Go to the Windows Store app and search for Ubuntu and install it.

```
$ sudo apt-get update

$ sudo apt-get install subversion ragel curl texinfo g++ bison flex cvs yasm
automake libtool autoconf gcc cmake git make pkg-config zlib1g-dev mercurial
unzip pax nasm gperf autogen bzip2 autoconf-archive p7zip-full python3-
distutils –y

$ mkdir ffmpeg

$ cd ffmpeg

ffmpeg $ git clone https://github.com/rdp/ffmpeg-windows-build-helpers.git

ffmpeg $ cd ffmpeg-windows-build-helpers

ffmpeg-windows-build-helpers $ sudo ./cross_compile_ffmpeg.sh –disable-
nonfree=n
```

Incoming Stream Handler

- Handle POST request
- Receive video data from external device
- Transmit video data every 1024 bytes

```
for {
    data, err := ioutil.ReadAll(io.LimitReader(r.Body, 1024))
    if err != nil || len(data) == 0 {
        break
    }

    s.clientManager.BroadcastData(&data)
}
```

WebSocket Handler: ServeWS()

```go
func (h *WebSocketHandler) ServeWS(w http.ResponseWriter, r *http.Request) {
   if r.Method != "GET" {
      http.Error(w, "Method not allowed", 405)
      return
   }

   ws, err := h.upgrader.Upgrade(w, r, nil)
   if err != nil {
      log.Println(err)
      return
   }

   log.Println("New client connected")
   client := NewClient(ws, h.unregister)

   h.register <- client

   go client.Run()
}
```

WebSocket Handler: Run()

```
for {
   select {
   case client := <-h.register:
      h.clients[client] = true
      log.Printf("New client registered. Total: %d\n", len(h.clients))
      break

   case client := <- h.unregister:
      _, ok := h.clients[client]
      if ok {
         delete(h.clients, client)
      }
      log.Printf("Client unregistered.   Total: %d\n", len(h.clients))
      break
   }
}
```

Client Handler: ReadHandler()

```go
func (c *Client) ReadHandler() {
   defer func() {
      c.unregisterChan <- c
   }()

   for {
      msgType, msg, err := c.ws.ReadMessage()
      if err != nil {
         break
      }

      if msgType == websocket.CloseMessage {
         break
      }

      log.Println("Received from client: " + string(msg))
   }
}
```

Client Handler: WriteHandler()

```go
func (c *Client) WriteHandler() {
   defer func() {
      c.unregisterChan <- c
   }()

   for {
      select {
      case data, ok := <- c.sendChan:
         if !ok {
            log.Println("Client send failed")
            c.ws.WriteMessage(websocket.CloseMessage, []byte{})
            return
         }

         c.ws.WriteMessage(websocket.BinaryMessage, *data)
      }
   }
}
```

Broadcast Video Stream

- IncomingStreamHandler call **WebSocketHandler.BroadcastData()**

```
func (h *WebSocketHandler) BroadcastData(data *[]byte) {
   for client := range h.clients {
      client.sendChan <- data
   }
}
```

- Client's WriteHandler

```
case data, ok := <- c.sendChan:
   if !ok {
      log.Println("Client send failed")
      c.ws.WriteMessage(websocket.CloseMessage, []byte{})
      return
   }

   c.ws.WriteMessage(websocket.BinaryMessage, *data)
}
```

JSMpeg Video Canvas

```
<span>Video</span><br/>
<canvas id="videoCanvas" width="1024" height="576">
   <p>
      Please use a browser that supports the Canvas Element, like
      <a href="http://www.google.com/chrome">Chrome</a>,
      <a href="http://www.mozilla.com/firefox/">Firefox</a>,
      <a href="http://www.apple.com/safari/">Safari</a> or Internet Explorer 10
   </p>
</canvas>
<script type="text/javascript" src="/static/jsmpeg.min.js"></script>
<script type="text/javascript">
       var url = 'ws://'+document.location.hostname+':8084/';
   var canvas = document.getElementById('videoCanvas');
   var player = new JSMpeg.Player(url, {canvas:canvas});
</script>
```

DEMO

- Setup FPV equipment
- **Run stream-server.go**

```
go run stream-server.go
```

- **Run ffmpeg**

```
ffmpeg.exe -f decklink -hwaccel cuvid -i "Intensity Shuttle" -c:v rawvideo -f mpegts -c:v
mpeg1video -b:v 800k -r 60  http://localhost:8082/secret
```

- **Access http://localhost:8080**

- **Repo.: https://github.com/chanshik/jsmpeg-stream-go**

References

https://github.com/chanshik/jsmpeg-stream-go

https://twitter.com/mrsteelefpv/status/858056393533923329

https://www.fatshark.com/product/hdo-fpv-goggles/

https://www.dji.com/kr/fpv

https://oscarliang.com/fpv-guide/

https://emaxmodel.com/tinyhawk.html

https://immersionrc.com/fpv-products/duo5800-5-8ghz-av-rx/

https://blackmagicdesign.com/products/intensity

Thank you for listening the presentation

Chan Shik Lim

chanshik@gmail.com