

## ABSTRACT

One of the greatest challenges with camera systems on High-Altitude Balloons (HAB) is to control the direction the camera points. This is because the payload boxes constantly rotate during the flight. While mechanical gimbals are normally used, an electronic gimbal was designed using multiple cameras and a series of multiplexers. The design consists of 8 version 1 raspberry pi cameras, connected to a Raspberry Pi through a series of 7 multiplexers that interface with an inertial measurement unit (IMU) to tell the direction the cameras are facing. This design attaches to the Raspberry Pi through the GPIO pins. The multiplexers turn on the camera that is facing the direction desired and send the video data from the camera to the Raspberry Pi, which then sends it to the ground station through a 5.8 GHz modem.

## OCTOPLEXER

The design consists of a network of 7 high speed differential 2 input 1 output FSA642 multiplexers connected to 8 version 1 Raspberry Pi cameras through ribbon cables. It also has a BN055 inertial measurement unit (IMU) to identify the orientation of the cameras. The network of multiplexers and the IMU are controlled by the atmega28p microcontroller using the I2C protocol. The functionality of the board in a big picture is the following:

1. -Initialize the Atmega328p
2. -Initialize the BN055 and calibrate the unit
3. -Select the correct multiplexers to have the right camera pointing in the direction desired and gather its data.
4. -Video Data is transferred to the Raspberry Pi for streaming.



```
void cam8(void){
  Serial.println("Switch to Camera 8");
  digitalWrite(SEL1,HIGH);
  digitalWrite(SEL2,HIGH);
  digitalWrite(SEL3,HIGH);
  state = 8;
}

//Data communication between ATMEGA328P and Raspberry Pi
void readDesired(void){
  int bytes1 = (digitalRead(Pi01)*8) + digitalRead(Pi02); // Decimal value that views like binary
  int data = (digitalRead(Pi01)*8) + (digitalRead(Pi02)*4) + (digitalRead(Pi03)*2) + digitalRead(Pi04);
}
```

Figure 1: Octoplexer board (left) and part of code to program the board (right)

## VIDEO TRANSMISSION

One of the main purposes of the video system is to capture video and stream it on the web live. To accomplish this, the payload requires a Ubiquity network that transmit data to the ground station through a 5.8 GHz modem. This system is automated so that after the payload is turned on, the Raspberry Pi will set up the network by grabbing a static IP. The Raspberry Pi, then detects the camera connected to it. Once, the camera is detected, the video starts by running a script that contains all the arguments of the video. The video data is then piped to Ffmpeg and copied into the SD card of the Raspberry Pi. The modem is the medium through which the data is transmitted from the payload mid flight to the ground station.

Before the transition to Ffmpeg, the video was streamed and piped using VLC. In the following table there is a comparison of VLC and Ffmpeg.

| FFMpeg  | VLC   |
|---|---|
| <ul style="list-style-type: none"> <li>• Unstable</li> <li>• Less delay</li> <li>• Configuration is challenging</li> <li>• Breaks with data gaps</li> </ul> | <ul style="list-style-type: none"> <li>• Stable</li> <li>• Slow</li> <li>• Easy configuration</li> <li>• Handles data gaps</li> </ul> |

Figure 2: Table Comparison Ffmpeg vs VLC

## DISCUSSION AND RESULTS

The video payload was tested multiple times on the ground as well as in flight. The multiplexer worked accordingly keeping the video looking in a chosen direction when the balloon was spinning. The delay from the video stream to the website was reduced. The only inconvenience was that whenever there was a gap in the video data coming from the payload, Ffmpeg would crash. The only solution to get it back was to restart Ffmpeg and re-run it. Frame rates were adjusted to get optimal streaming given the low bandwidth of some internet connections.

The design of the Raspberry Pi camera changed during the course of the project resulting in the octoplexer only being usable with version 1 Raspberry Pi cameras. Later versions of the Raspberry Pi cameras have an encryption IC allowing only one camera to work with that specific Raspberry Pi.

## FUTURE WORK

For the future, one of the things to add to the system is to find a way to use Raspberry Pi cameras version 2 with the octoplexer. Also, add a feature into the ground station GUI to restart Ffmpeg automatically in case it crashes. Another thing to work on, is to display information like altitude, temperature and pressure in the live video feed.

## ACKNOWLEDGEMENTS

- Montana Space Grant Consortium
- Stream.Live