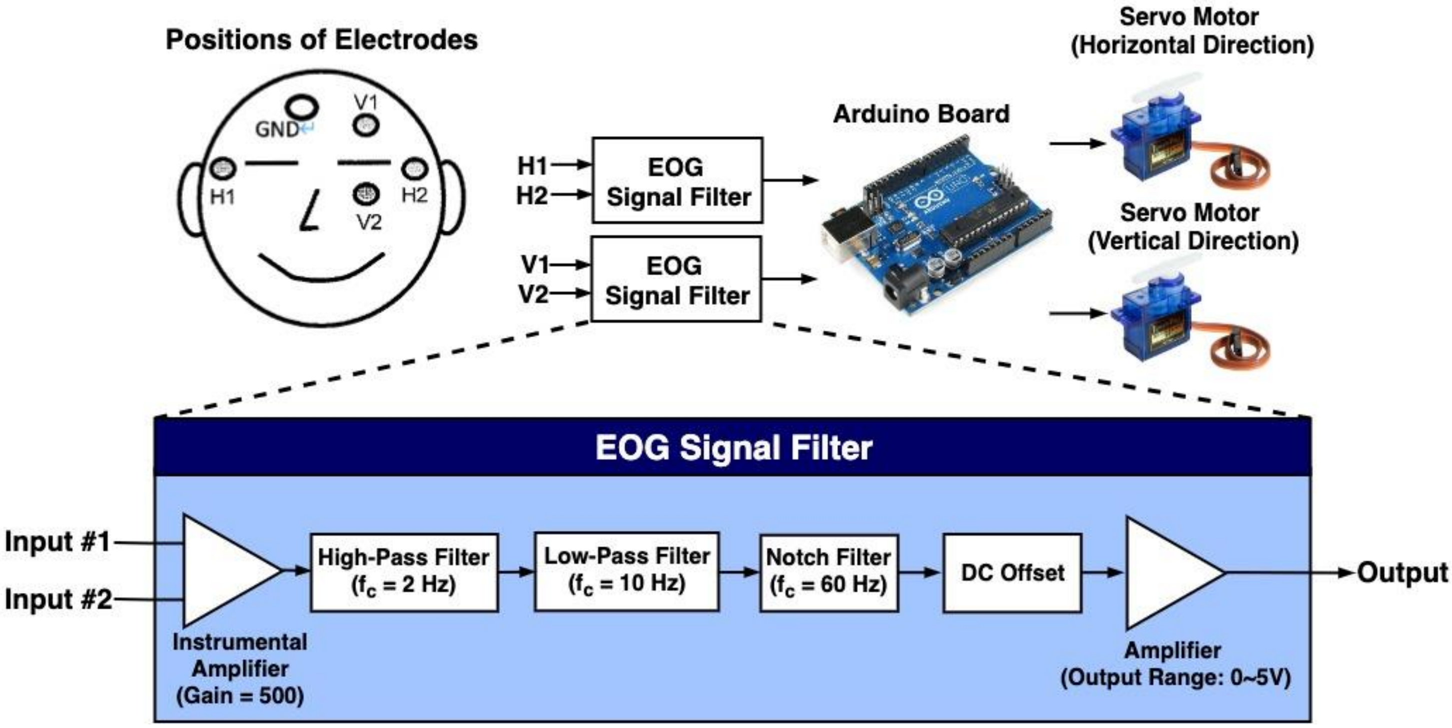
Webcam Controlled by Electrooculography (EOG) Signal

Circuit Design

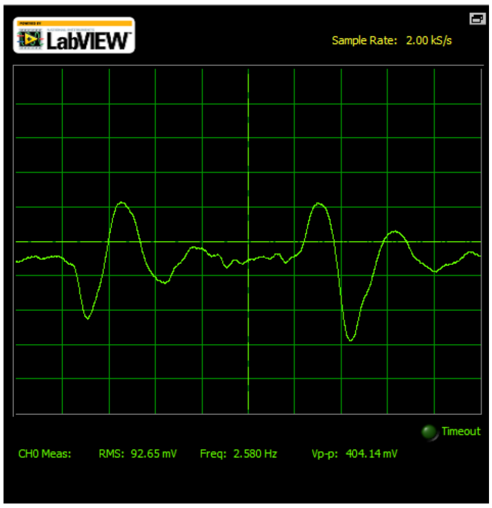
Searching information from Internet, we discovered that EOG signal's bandwidth is between 2Hz and 10Hz. The signal's voltage value is from 1 to 10 μV. To process the signal easily, we applied the instrumental amplifier to amplify 500 times the EOG signal. Then, the signal passed through the high-pass filter and the low-pass filter whose cutoff frequency are 2Hz and 10Hz respectively. Afterwards, a notch filter removed the noise whose frequency is 60Hz. Before inputting the signal to the Arduino board, the DC voltage source was modified to make the EOG signal be between 0 and 5 V. Lastly, the Arduino board started to analyze the signal and controlled the servo motors which were responsible for webcam's movement.


Positions of Electrodes
GND
V1
H1
H2
V2
H1
H2
EOG Signal Filter
V1
V2
EOG Signal Filter
Arduino Board
Servo Motor (Horizontal Direction)
Servo Motor (Vertical Direction)
EOG Signal Filter
Input #1
Input #2
Instrumental Amplifier (Gain = 500)
High-Pass Filter (f_c = 2 Hz)
Low-Pass Filter (f_c = 10 Hz)
Notch Filter (f_c = 60 Hz)
DC Offset
Amplifier (Output Range: 0~5V)
Output

We implemented two filter circuits for vertical and horizontal movement. There were slight difference in the parts of the DC offset and the final amplifier. The rest of the circuits are the same.

After the measurement, we found that the differentiation between the signals for upward and downward direction. However, it is obvious that the signals related to moving right and left were the inversion of each other.

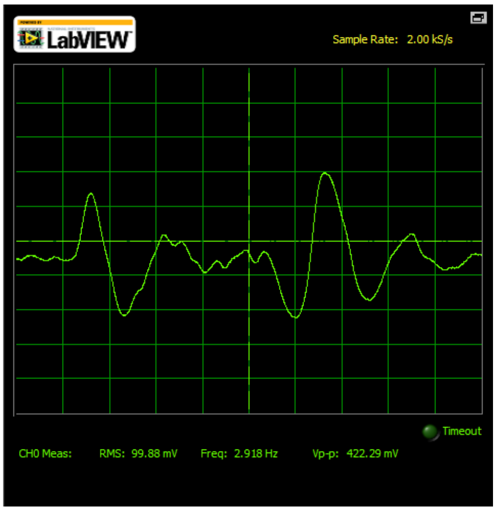
Horizontal movement signals



Sample Rate: 2.00 kS/s

CH0 Meas: RMS: 92.65 mV Freq: 2.580 Hz Vp-p: 404.14 mV

Left

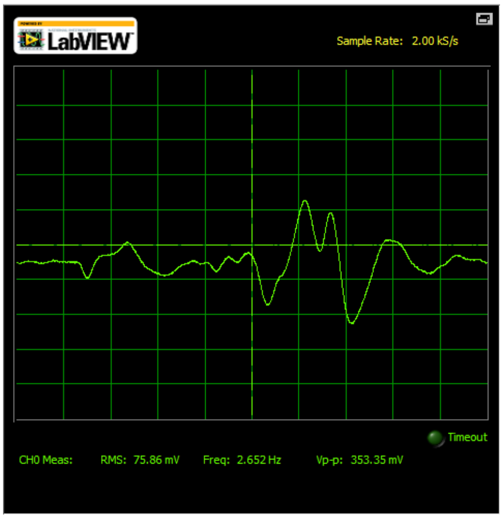


Sample Rate: 2.00 kS/s

CH0 Meas: RMS: 99.88 mV Freq: 2.918 Hz Vp-p: 422.29 mV

Right

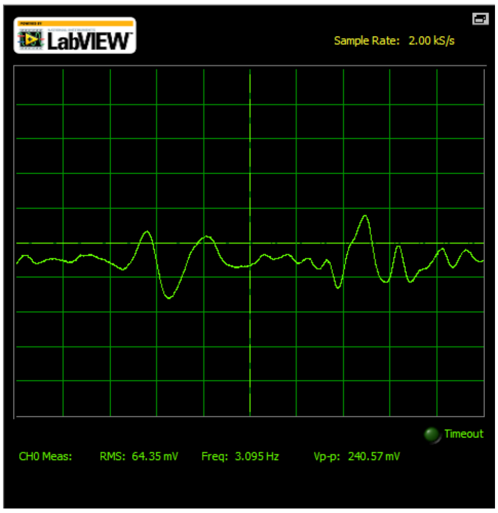
Vertical movement signals



Sample Rate: 2.00 kS/s

CH0 Meas: RMS: 75.86 mV Freq: 2.652 Hz Vp-p: 353.35 mV

Up



Sample Rate: 2.00 kS/s

CH0 Meas: RMS: 64.35 mV Freq: 3.095 Hz Vp-p: 240.57 mV

Down

In the figures, the signal for downward movement seems like the noise. We re-built the circuit again and the signal quality was improved. Nevertheless, the signal still could not be processed by the Arduino board to control the servo motor for vertical movement correctly. The board could only accurately manipulate the servo motor for horizontal movement.

Webcam Video

The webcam was connected to the Arduino Yun board. Relative programs were installed to the Arduino Yun board with the Terminal. The Arduino Yun board was used as a live stream server. It could upload the video to the website in real-time.

After finishing the connection of the Internet and the Arduino Yun Board, open the Terminal and use the command "ssh" to connect to the board.

```
sudo ssh root@arduino.local
```

If log in, install the driver for the webcam.

```
opkg update

opkg install kmod-video-uvc

opkg install fswebcam
```

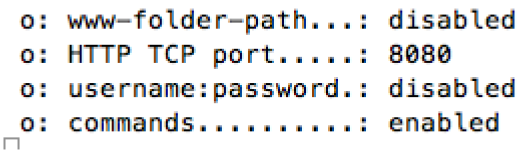
Install mjpg-streamer for image acquisition and live stream server.

```
cd /www

mkdir webcam

cd /www/webcam

opkg install mjpg-streamer
```

Activate mjpg-streamer. The number after "-f" means fps.

```
mjpg_streamer -i "input_uvc.so -f 20 -d /dev/video0" -o
"output_http.so"
```

If the setting is complete, the following message will show up:

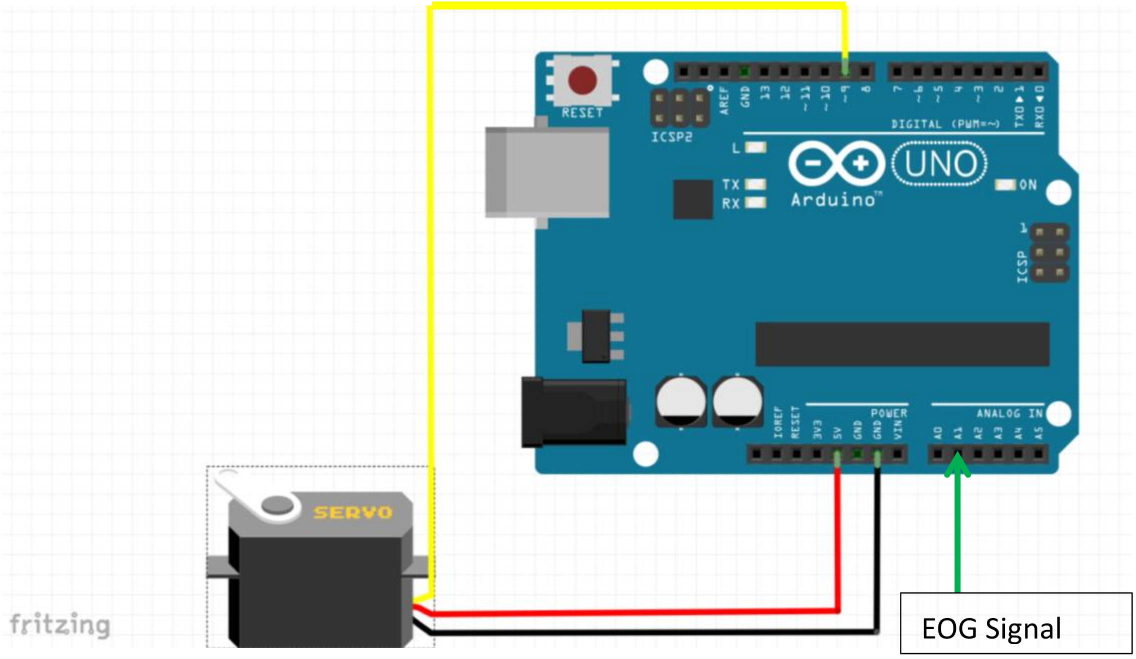


```
 o: www-folder-path...: disabled
 o: HTTP TCP port.....: 8080
 o: username:password.: disabled
 o: commands..........: enabled
```

Open the browser and enter
`http://arduino.local:8080/?action=stream`
The video captured by the webcam will be presented on the website.

Not all webcam are supported by kmod-video-uvc (please check http://www.ideasonboard.org/uvc/ for more details). Hence, we used Logitech C310 webcam which is supported by the software package in our final project.

EOG Signal Controlling Servo Motors



After the circuit finished filtering the EOG signal, the signal is adjusted to 0-5V to avoid overload of the Arduino board's analog input pin's overload. When the signal was transferred to the board, we use Serial Port Monitor to check EOG signal's value and set up the upper and lower bound or the signal. The function was similar to the comparator. If the input signal was larger than the upper bound, the Arduino board would receive the value for upper bound. On the other hand, the board would be given the value for lower bound if the signal was less than the lower bound. By this function, we could analyze the waveform of EOG signals and get the characteristics, such as the direction, the duration of fixing eyes to a direction, and the time when the motor stop moving.

The webcam was attached to the upper part of the pan-tilt kit. The servo motors were installed at the bottom of the pan-tilt kit to control the movement of the webcam.

Pan-Tilt Kit

Arduino Code

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <Servo.h>
// Initialize motors, upper and lower bound, motors' initial angles, and input analog signal

Servo Par_Servo, Ver_Servo;
int lowLimit = 550, highLimit = 820;
int parPos = 90;
int Par_SensVal;
bool delayFirstTime = true;

void setup() {
	// Set Serial Port Monitor, the pin for controlling the motors, and motors' initial angles

	Serial.begin(9600);
	Par_Servo.attach(9);
	Par_Servo.write(parPos);
}

void loop() {
	char str[256];
	if(delayFirstTime)		// Exclude the unwanted signal
	{
		delayFirstTime = false;
		delay(1000);
	}
	else
		delay(100);

	// Analyzing EOG signal (Find the value for upper and lower bound)
	while(true)
	{
		Par_SensVal = analogRead(A1);
		Par_SensVal = constrain(Par_SensVal, lowLimit, highLimit);
```

```
      sprintf(str, "Par_SensVal = %d ", Par_SensVal);
      Serial.println(str);
      if((Par_SensVal == lowLimit || Par_SensVal == highLimit))
      break;
  }
  // If the value is less than the lower bound, enter the part for turning left
  if(Par_SensVal == lowLimit)          // Turn left
  {
      // Analyze the waveform and check whether it is upward and surpass the upper bound
      while(true)
      {
          Par_SensVal = analogRead(A1);
          Par_SensVal = constrain(Par_SensVal, lowLimit, highLimit);
          sprintf(str, "Par_SensVal = %d ", Par_SensVal);
          Serial.println(str);
          if(Par_SensVal == highLimit)
              break;
      }

      // Check whether the waveform enters the middle and smooth part
      while(true)
      {
          Par_SensVal = analogRead(A1);
          Par_SensVal = constrain(Par_SensVal, lowLimit, highLimit);
          sprintf(str, "Par_SensVal = %d ", Par_SensVal);
          Serial.println(str);
          if(Par_SensVal != lowLimit && Par_SensVal != highLimit)
              break;
      }
      // Motor spins left
      for(;parPos < 180; parPos++)
      {
        Par_SensVal = analogRead(A1);
        Par_SensVal = constrain(Par_SensVal, lowLimit, highLimit);
        sprintf(str, "Par_SensVal = %d ", Par_SensVal);
        Serial.println(str);

        // The signal is less than lower bound, and the motor stops spinning
        if(Par_SensVal == lowLimit)
            break;
        Par_Servo.write(parPos);
        delay(10);
      }
}

// If the value is larger than the upper bound, enter the part for turning right
else if(Par_SensVal == highLimit) // Turn right
{
    // Analyze the waveform and check whether it is downward and surpass the lower bound
    while(true)
    {
        Par_SensVal = analogRead(A1);
        Par_SensVal = constrain(Par_SensVal, lowLimit, highLimit);
        sprintf(str, "Par_SensVal = %d ", Par_SensVal);
        Serial.println(str);
        if(Par_SensVal == lowLimit)
            break;
    }
```

```
// Check whether the waveform enters the middle and smooth part
while(true)
{
        Par_SensVal = analogRead(A1);
        Par_SensVal = constrain(Par_SensVal, lowLimit, highLimit);
        sprintf(str, "Par_SensVal = %d ", Par_SensVal);
        Serial.println(str);
        if(Par_SensVal != lowLimit && Par_SensVal != highLimit)
                break;
}
// Motor spins right
for(;parPos > 0; parPos--)
{
        Par_SensVal = analogRead(A1);
        Par_SensVal = constrain(Par_SensVal, lowLimit, highLimit);
        sprintf(str, "Par_SensVal = %d ", Par_SensVal);
        Serial.println(str);
        // The signal is larger than the upper bound, and the motor stops spinning
        if(Par_SensVal == highLimit)
                break;
        Par_Servo.write(parPos);
        delay(10);
}}}
```

Reference

Arduino Yun Webcam： http://oranwind.org/arduino-arduino-yun-webcam-zi-zhi-jian-shi-qi/