1. Had to use 3 to 2 prong adaptor (to disconnect earth/gnd wire) on light organ power plug to avoid servo noise getting on the audio speakers as well as falsely triggering the light organ lights. Need to check inside light organ to see what that gnd is connected to as it appears noise is getting on this gnd/earth wire which gets back thru the circuity to the audio line in and then back into the audio speakers. Also the jack on the light organ was a little flakey and required the plug to be slight pulled out to work so need to work on that too (just need to verify continuity of the jack wires inside the light organ to the other end of the audio plug when it is plugged in).   
   Ensure that the volume level control of the PC is set appropriately otherwise if it is not high enough the light organ won’t work. The organ has gain sliders on it as well so also check those although they are typically left in the same positions used on Halloween. Used the PC 65% volume level to get the organ to work at its current settings. Used the LL and MM outputs of the organ and that seemed to work ok.
2. Eventually to eliminate position recordings size limits, see if can get an SD card attachment for the Arduino so can store them no there. Not a big deal at the moment as can store all the current recordings (50msec interval) and total program memory taken up is 97K out of 256K.
3. Didn’t use the air cylinder again this year so not sure if that’s working ok still.
4. Replaced bottom half of the eye LEDs connection wire from the head with the 3 line servo wire to make it more flexible.
5. Check to see if can use the aluminum servo hub in place of the plastic one on the mouth servo although the plastic one seems to work ok.
6. See if can get some swing point on the left side of the mouth connected to the head to provide more support as didn’t use the backup/down since adding the mouth servo. Not sure if back up/down will cause problems with this so need to try out.
7. Ensure Foscam is working ok with laptop a few days before (use Internet explorer tab in chrome to access all functionality)
8. Now have ability to attach IMUs to head and mouth so that prop mimics my movements. Head IMU attached with duct tape to the top of a baseball cap (see orientation markings on the IMU). Mouth IMU is permanently attached to a chin guard with an elastic strap to hold it onto my head. Adjusted the software scaling for the mouth movements to optimize it so prop mouth opens wider than normal mouth does.  
   The up/down side of the IMUs is important in that if an IMU is placed upside down a surface (the chip side of the board is the up side , i.e pointing up, the solder side is the down side ) it can cause the yaw (heading) readings to go haywire when the pitch of the IMU is outside a certain range. If upside down, then when keeping the heading direction the IMU is pointing constant and moving the IMU so the pitch starts to change beyond a certain range (more than 30 degrees from horizontal pointing down) the IMU heading reading suddenly jumps to large values like 255 or greater. There may be an orientation setting that could solve this but simply having the IMU upside pointing up fixed the problem.
9. Do more mouth scaling to avoid having to exaggerate my mouth movements to get reasonable prop mouth movements.
10. Now have a way see, talk, and hear on prop, so can do live interactive prop operation. This seemed to work fine buy may have one issue described below. Used mini Relohas wifi cam on top of prop head to see, and Electret Microphone Amplifier - MAX9814 with Auto Gain Control to hear what’s going on (to headset earphones with volume ctrl), and headset mic connected to and replacing mic on Electret Microphone Amplifier - MAX9814 with Auto Gain Control to existing prop speaker for talking and IMUs on head gear for head and mouth movement. Electret Microphone Amplifier - MAX9814 with Auto Gain Control device eliminates any audio delay and delay between audio and mouth movements. Using laptop caused too much delay in audio. Relohas mini cam has very little video delay so that worked well (uses smartphone to view live video)  
    **For the Mic to prop speaker the use of the MAX9814 with AGC to amplify the mic looked to be picking up sounds quite a distance away from the mic due to the AGC so see about using a non AGC amplifier for this from Adafruit to eliminate this.** The MAX9814 with AGC worked well for the ears of the prop as it picked up very faint as well as very load sounds because of the AGC.
11. **Remember to connect mic audio**. In 2020 had forgotten to do this for the first few TOTers and they couldn’t hear me talk.
12. See if there is some way to make a skin cover for the bottom of the mouth to the neck to cover up the servos and make it look better.
13. See if there is a way to limit the head from completely bending over where the chin hits the neck pvc pipe, when the servos are not powered. Possibly use a spring that has a little give and then attach it along with some slack wire to the center of the vertical servo metal connection so that when that servo is holding the head at 160 degrees or 20 degrees the slack/cable is completely taught and starts to pull on the spring. If the spring has a little give it can cushion the head when it drops from 90 degrees to those positions. As a work around set head vertical to 50 degrees before disconnecting power so that had falls backwards.
14. Have the servo device class have a method that limits the servo setting to provide a better way to limit this in one place (possibly in the base device class so every one inherits it)
15. See if can also have the prop audio go to the loud speakers (although the prop speaker did seem to be loud enough)
16. Have light shining on the vampire prop itself to make it more visible as people can stand in the way of the porch light that is the only light on it now.
17. Last time used base had a problem trying to fit the prop into it and ended up not using it? Will need this working if want to use back up/down.
18. See about adding further quick connect/disconnect right at controller board for lines that go to prop and to IMU head gear to make it easier to transport board (with the exception of the thick yard cable although would be nice for that too)
19. Did add blocking capacitor to receive audio from prop before it goes to the Headset headphones. That helped overdrive of speaker when headset volume control turned way up
20. Would using quaternions instead of Euler angles avoid issues for the planned future arm movement. Hopefully the computations will not overload the processor. Right now it’s not an issue due to the limits of head movements but adding arm movements may run into issues. Also since arms physical motor connections (rotate, altitude, roll) would be setup as yaw, pitch, roll anyway would we still have the issue regardless? Need to investigate more. There is code that has special case when pitch is 90 degrees so that roll is always set as 0 and yaw works as is. Maybe that would work well enough. The basic problem is at certain points of the arm position (pitch 90 deg), moving just a little off the position can cause the yaw, roll to change drastically to move to this new slightly different position because yaw and roll do the same thing at pitch 90 deg.

Enhancements -

1. Making the prop skeleton arm move with servos : one for the hand, up/down two for the lower part of the arm (up/down and twist) and three for the upper arm part (Vertical up/down (pitch), Horizontal rotate (yaw) , and Roll) where the upper part would need to be geared to get the force needed to lift the arm or a really strong servo. Prefer strong servo as it will move quicker. Want the arm to move at a decent natural speed. Would need another 3 IMUs for choreographing for this (already have one but having a spare is always a good idea) to allow arm shoulder up/down, arm shoulder side to side, and forearm up/down (a fourth for hand up/down or forearm rotate around its axis)It appears that should be able to have another 9 servos that the Arudino Mega can control by using 16 bit Timers 3,4,5 (3 servos for each timer and timer 1 already used for the 3 existing servos)) which currently no other arduino mega packages I use actually use, or plan on using use, as far as I can tell by searches. See the Atmega2560 datasheet for any difference in using those from timer 1 (17.2 Overview and on in that datasheet). Would need to modify the PWMServo package to use these additional timers. Optionally there is a adafruit shield that uses I2C that can control a massive number of servos ( it is 14 bit instead of 16 bit but that should be good enough for 1 degree changes). I2C should be fast enough even to command 9 servos every 50 milliseconds. Note that if use same IMUs currently using would need an I2C hub switch to implement this as the IMUs can only have 2 I2C IDs  
   **Note that if using PWM from timer3,4,5 get a 1.64volt very short noise glitch on the pwm output lines unless use the 2nd ground from the Arduino. So need to look at shortening ground wire lengths (was using ground on audio board to the scope that seemed to result in that glitch, but it went away when used gnd right at the Arduino for looking at pwm on the scope**. May end up using a separate servo shield with I2C interface at the prop to reduce the additional lines from the Arduino to the prop. Only need the I2C lines , gnd, vcc, and the servo power. How the servo power is feed to the servos remains TBD as would like to still be able to disconnect power servo by servo or at least by groups of servos. Possibly can feed some of the power by the existing 3 servo power lines to the prop and gang up more than 1 servo on a given power line possibly using a bigger power supply. The other option is to have the power supplies at the prop and use relays and an I2C gpio board to switch power to the servos on and off. Or maybe just have extension cords for each servo or group of servos that can disconnect inside.  
   So I'm attempting it without using any torque multiplying servo gearbox's at first because I want the movement as fast as possible. For the shoulder I plan on using two relatively in expensive GoBILDA 2000 Series Dual Mode Analog Servo (25-2) (300 oz-in stall torque at 6Volts with no load servo speed of 0.2 sec/60degrees, weight 2.12 oz) for roll and yaw and a Savox SA1230SG Coreless Digital Servo (499.9 oz-in at 6 Volts, with no load speed 0.16 sec/60 degree, weight 2.8 oz) for pitch (i.e. lifting the arm up and down). Then another GoBILDA 2000 Series Dual Mode Analog Servo (25-2) for the elbow. My skeleton arm weighs 6.4 oz and is 24" long with its balance point around the 12" mark from the shoulder. So based on that I believe the bare min torque to hold the arm still straight out perpendicular to the body is (6.4 + 2.12) oz \* 12" = 102 oz-in of torque (assuming the elbow motor is right at the elbow although I could put it at the shoulder and use a light weight linkage to the elbow). So I figure a stall torque of say 3 times that or more should be good enough to move the arm at a reasonable rate and decelerate it ok and not overstress the servo. I suppose I could calculate the speed assuming the motor stall torque and arm weight distribution but I'm guessing 3 times the bare min still arm torque is good enough . My 499 oz-in motor is basically 5 times the bare min still arm torque. Most of the time the arm will be in the resting positing hanging straight down.
2. See if using some type of motor instead of the air cylinder would work for back up/down as it would be easier to control the stopping speed than the elastic cord mechanism we have now with the air cylinder. The only issue would be the speed it could achieve and the toque it can produce (has to be at least as quick as the pneumatic cylinder). Or if there is a way to more accurately slow down the pneumatic cylinder at the end points. Or could use proportional valve controllers or possibly servos hooked to valve controller and Arduino PIDs (proportional integration derivative) control using feedback of the position the cylinder is at (haunt forum has posts on this in the pneumatics forum)
3. Update code so that player registered devices override interactive prop device movement when player is playing. This allows playing short mouth and audio sequences during interactive prop movements (like a scream from the mouth).
4. See about having voice activated mouth control option. Purchased an audio level board, see if can use amp output of it in some manner instead of connecting off LEDs on that audio level board for audio level. The audio needs to have enough release to hold the level so that the Arduino is not taxed sampling that audio level.
5. Make LED eye brightness controllable. Either use DtoA output or pulsed width modulation.

2019 – 39 degrees high 22 low, snow on ground but not sidewalks - Count 60 Trick or Treaters

2020 – 58 degrees on Halloween night, clear and windy. Short of wind it was very nice night. Count 40 TOTers. Had candy in bags on table and used vampire prop to interact with TOTers only.