**PROTOTYPE DESIGN**

**Notes/Issues**

* Using Zoom API because it is available, supported, and likely platform for one-on-one video call
* How will Arduino/Python control webcam?
* Format for Arduino receiving video signal? Receive as hexadecimal or as binary and then convert?
* Protocol for video data transmission? Stream into a buffer?
* Need a figure/flowchart for prototype architecture

**Use Case:** Position webcam behind translucent screen to look through screen. In an alternating manner, webcam records user and screen displays partner. In this way, user makes eye contact with partner on screen without partner seeing their own reflection.

**Architecture**

* Hardware
  + USB Seeeduino v4.3
  + USB webcam
  + TOLED screen connected directly to Seeeduino
* Firmware
  + Arduino
  + Receives formatted video signal
  + Sends webcam control command (on/off)
  + Rapidly alternates between webcam and video display
* Software
  + Python
  + Runs on host PC
  + Pulls video signal from a source (e.g. Zoom API)
  + Formats video signal for TOLED
    - Black and white, adjustable threshold
    - 57 x 138 resolution
    - Region of interest, adjustable
    - Converts to hexadecimal?
  + Sends formatted video to Hardware via USB

**Milestones**

* Pull image from video and display on host PC (Python)
* Send formatted video to Seeeduino and display
* Alternate between displaying video and using webcam (no TOLED image on webcam feed)

**Tasks**

* Python
  + Format image from video
    - Open and display image
    - Threshold to black and white
    - 56x128 resolution (scale and/or ROI)
    - Display new image
    - Convert image to hexadecimal code?
  + Identify video and camera controls in Zoom API
  + Grab video
    - Open video feed from webcam (via Zoom API)
    - Sample image from video feed
  + Send image data to Seeeduino for display
* Arduino
  + Understand hexadecimal display code
  + Read image data via USB to display image (as hex, or convert from binary?)
  + Rapidly swap between camera feed and video feed