Robotic Makeup Application System

Alejandro Varela
Dillon Naidu
Jason Ramsarran
Jason Calle
Vincent Xu
Alex Wong

Department of Mechanical Engineering
The City College of New York

Requirements

- Collaborative
- Responds and interacts with the user
- LCD display and buttons
- Motion sensor

Trade study: Robotic arms

Universal Robots UR3

uArm Swift Pro

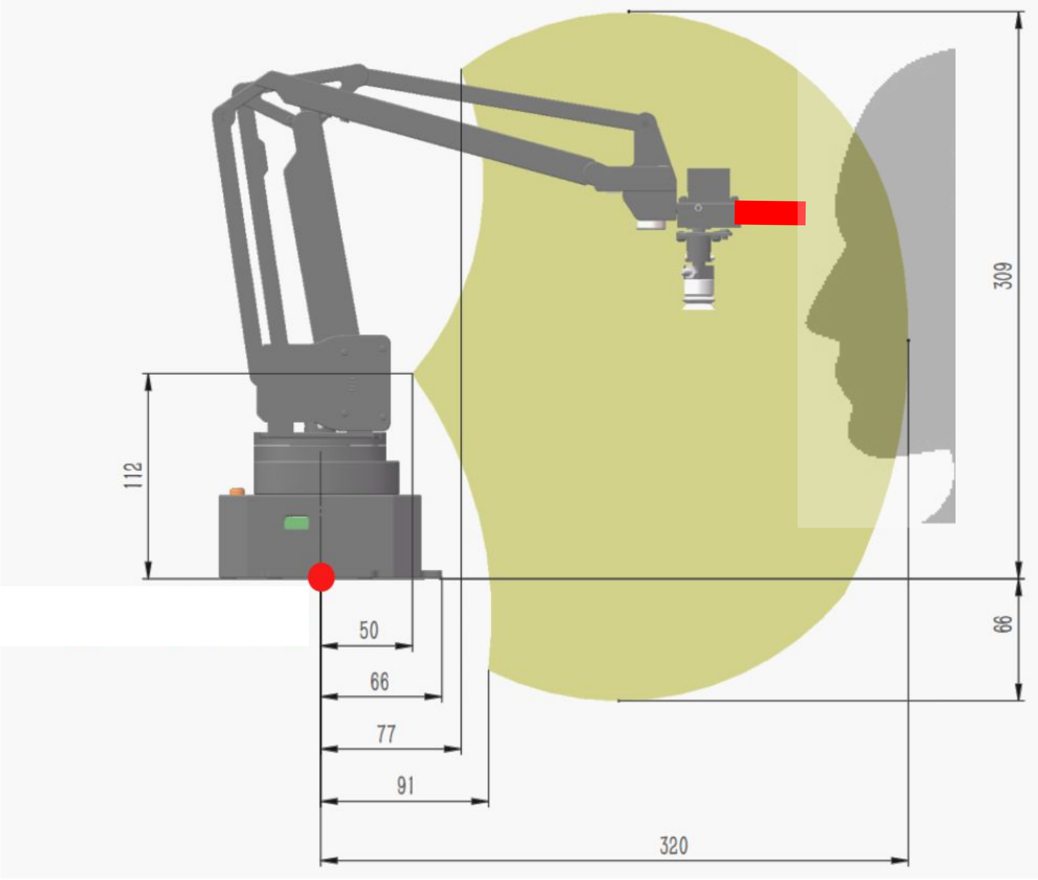
Braccio

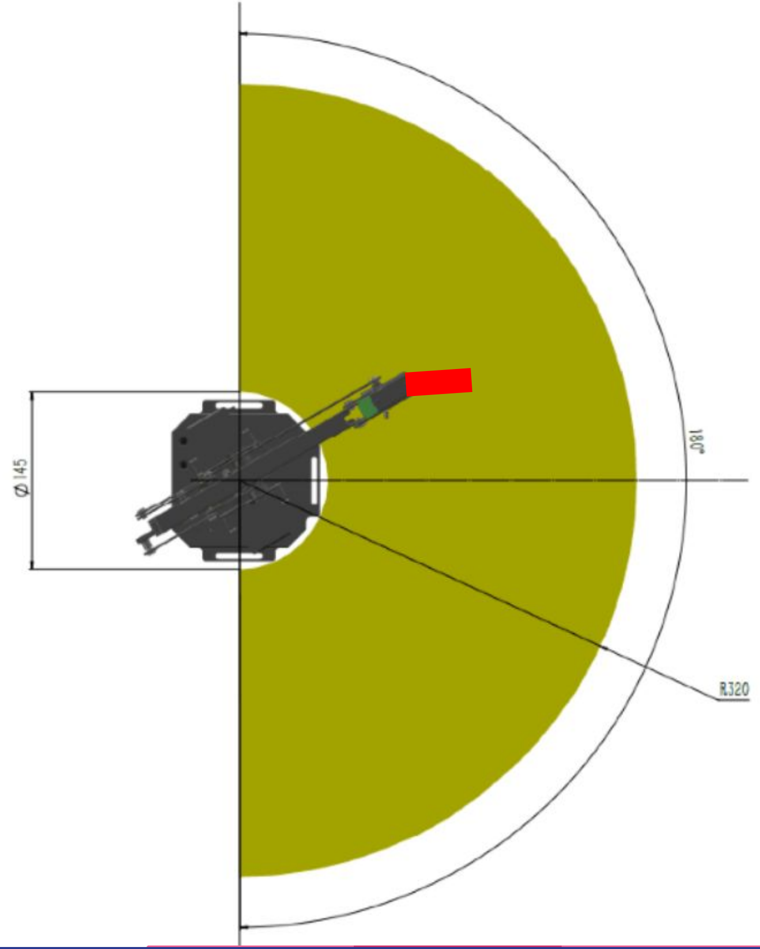
Trade study: Robotic arms

Given scores of 0-3 (higher is better)

Weight	3	3	3	2	2	2	2	2
Robotic Arm	**Affordability**	**On-board software**	**User-friendly**	**Degrees of Freedom**	**Repeatability**	**Expandability**	**Appearance**	**Compactness**
uArm Swift Pro	2	3	3	2	3	3	3	3
Universal Robots UR3	0	3	3	3	3	2	3	0
Braccio	3	1	2	3	1	1	2	2
Weighted score:	**uArm Swift Pro**	Universal Robots UR3	Braccio					
	41	40	36					

uArm Swift Pro



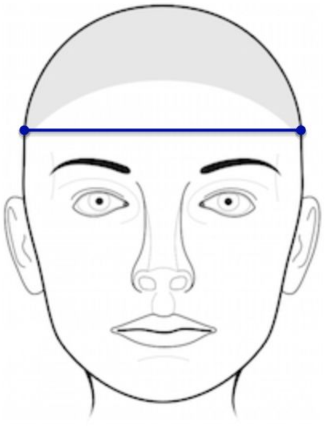


Face Dimension Statistics

Cranial Width

Average Male - 150 mm

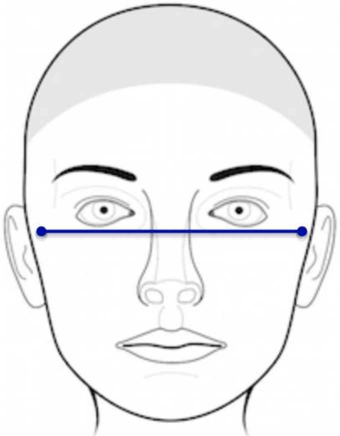
Average Female - 145 mm



Facial Width

Average Male - 135 mm

Average Female - 130 mm

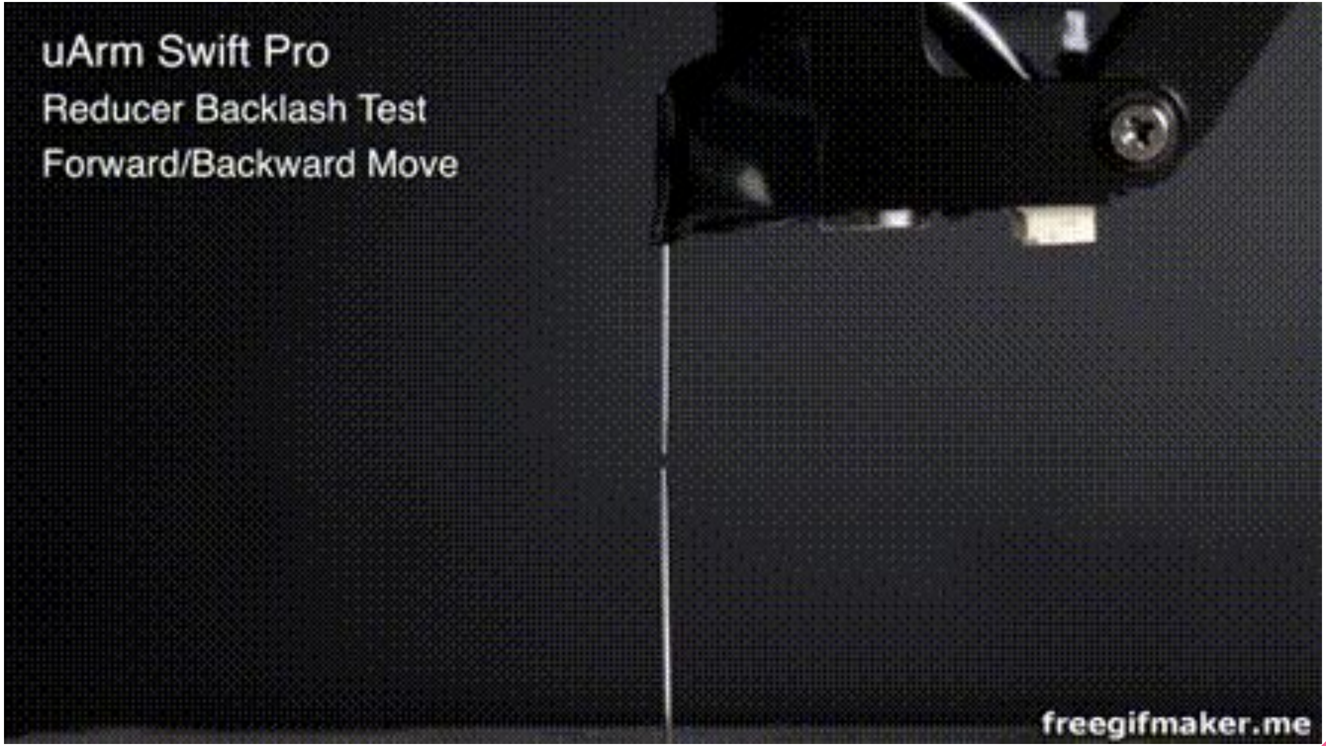


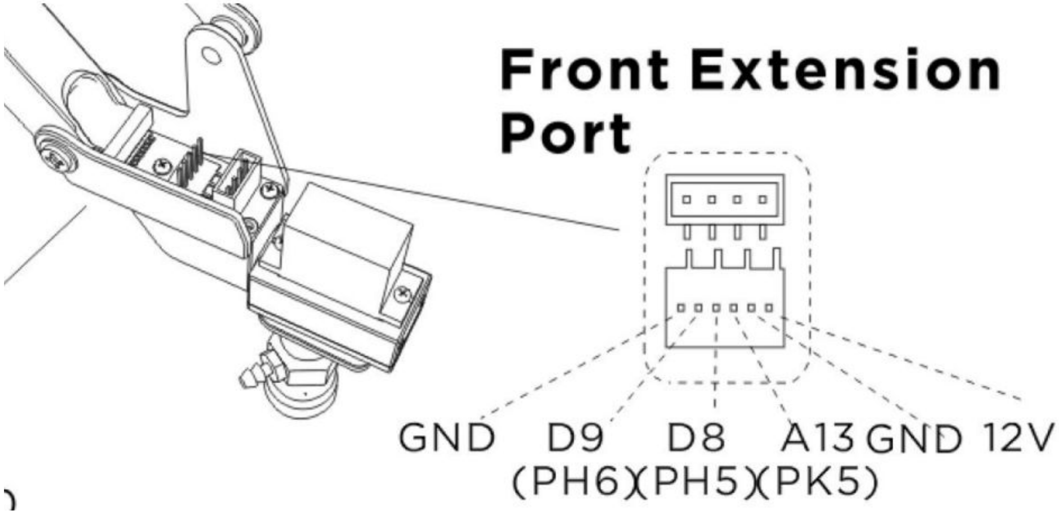
Facial Length

Average Male - 203 mm

Average Female - 190 mm

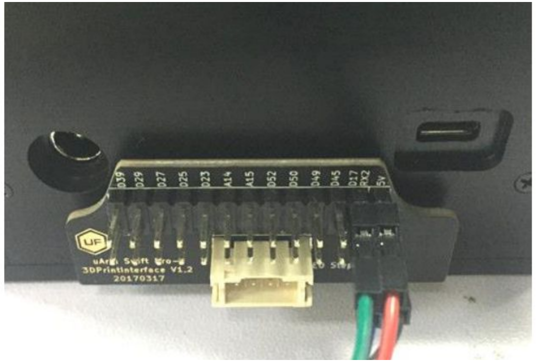
Specifications	
	uArm Swift Pro
Weight	2.2kg
Degrees of Freedom	4
Repeatability	0.2mm
Max. Payload	500g
Working Range	50mm ~ 320mm
Max. Speed	100mm/s
Connector	Micro USB
Wireless	Bluetooth 4.0
Input Voltage	DC 12V
Power Adapter	Input:100 ~ 240V 50/60Hz; Output: 12V5A 60W

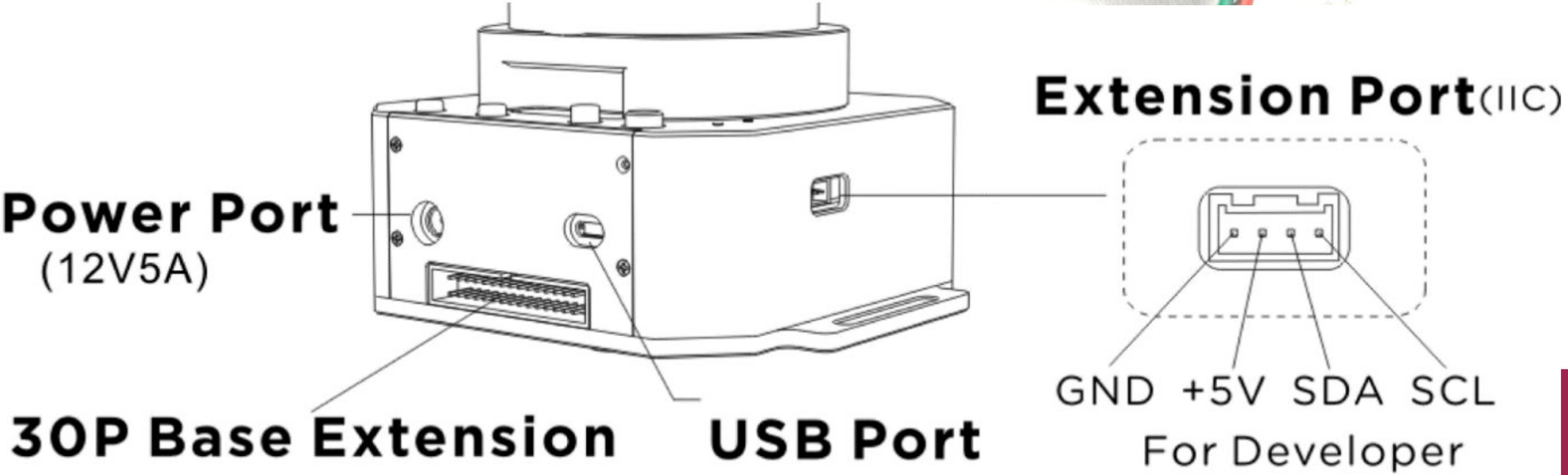

uArm Swift Pro
Reducer Backlash Test
Forward/Backward Move
freegifmaker.me


Front Extension Port
GND
D9 (PH6)
D8 (PH5)
A13 (PK5)
GND
12V


UARM
FRIXION BALL

Extensions




Extension Port(IIC)
Power Port
(12V5A)
GND +5V SDA SCL
For Developer
30P Base Extension
USB Port

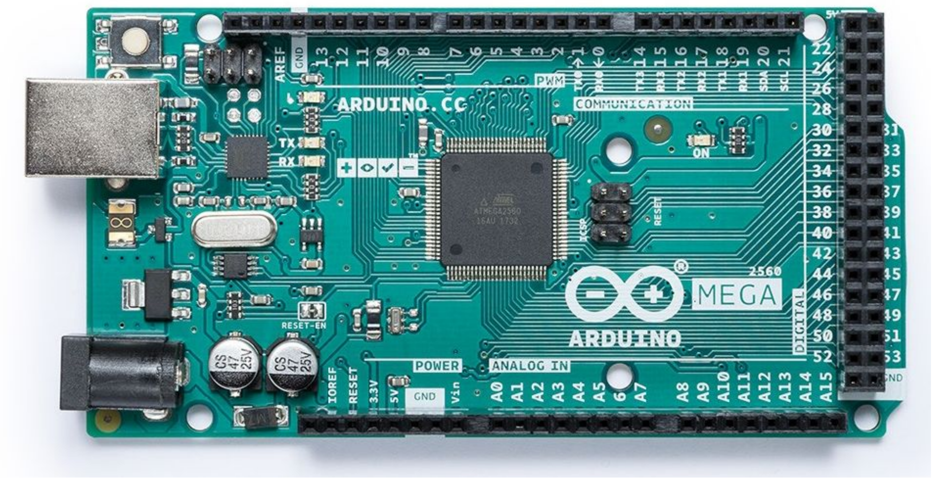
Trade study: Microcontroller

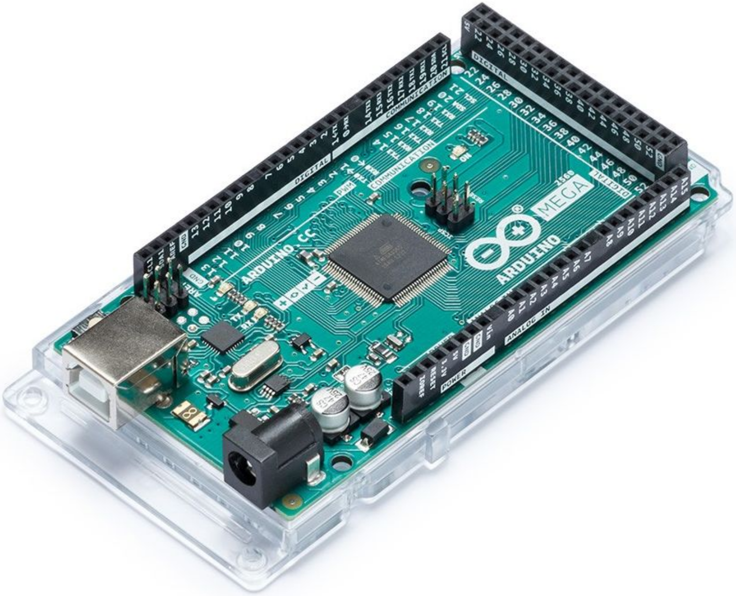
Given scores of 1-3 (higher is better)

Weight	2	2	1	1
Type of Microcontroller	**I/O performance**	**CPU Intensive Performance**	**Web Support and Example Projects**	**Compatibility with robotic arm**
Raspberry Pi	1	2	3	2
Arduino	3	2	3	3
Weighted score	**Raspberry Pi**	**Arduino**		
	11	**16**		

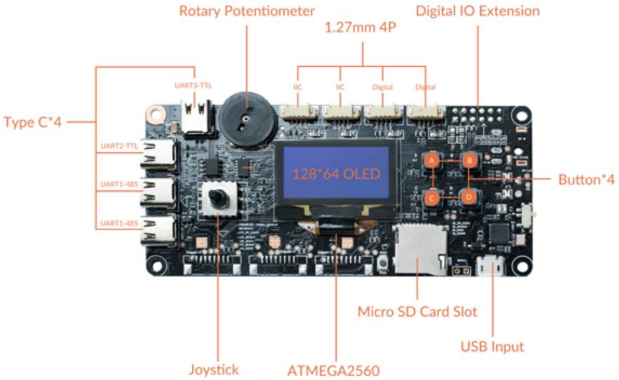
Arduino Mega 2560

- 54 digital I/O pins
- 16 analog inputs
- large space for sketches in Arduino IDE

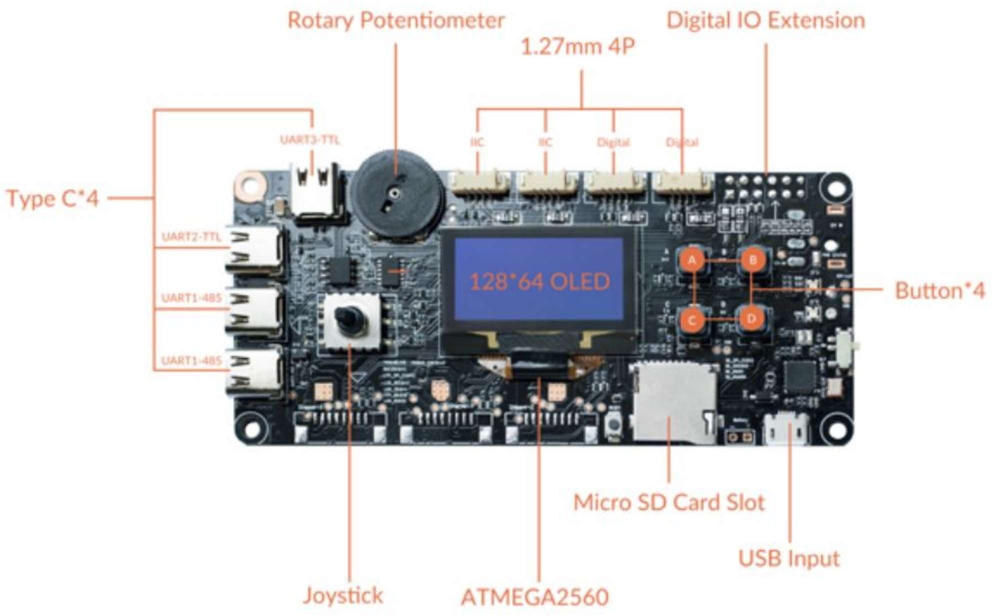




Hardware Pictures



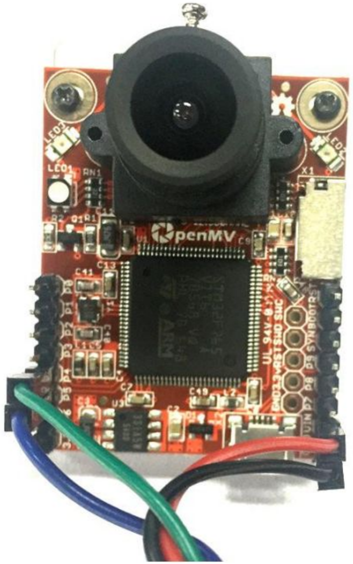
uArm Arduino Controller

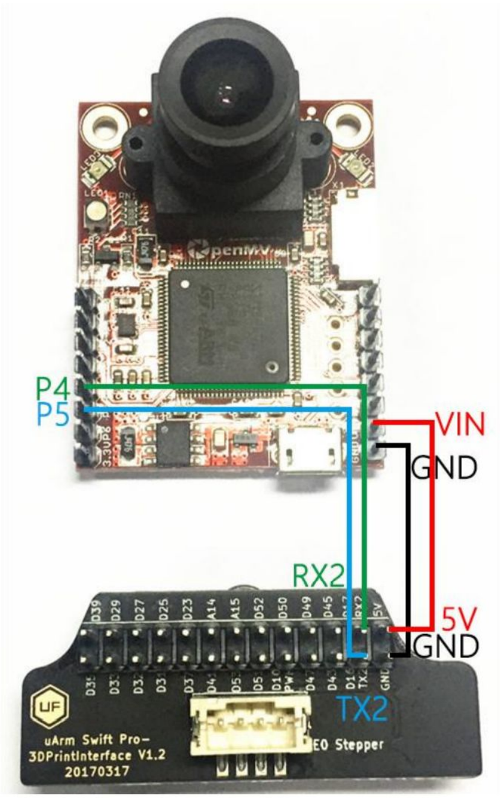


- Plays collaborative role
- Receives information from all components
- Programming flexibility

OpenMV Cam

uArm Swift Pro with extended port to connect with sensor





OpenMv Cam with uArm Swift Pro

Example of face detection:

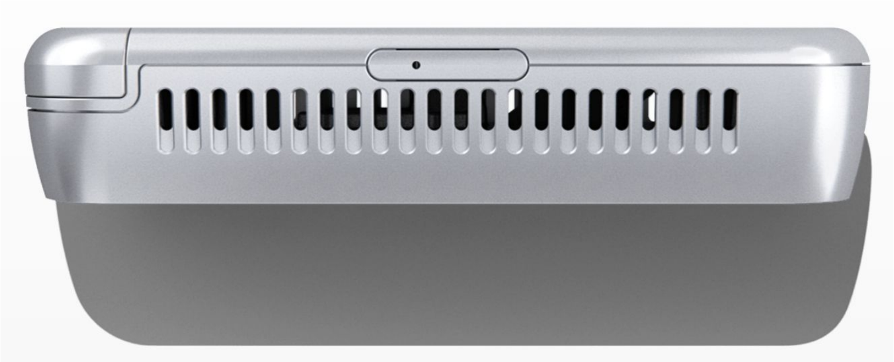
Trade study: Sensors

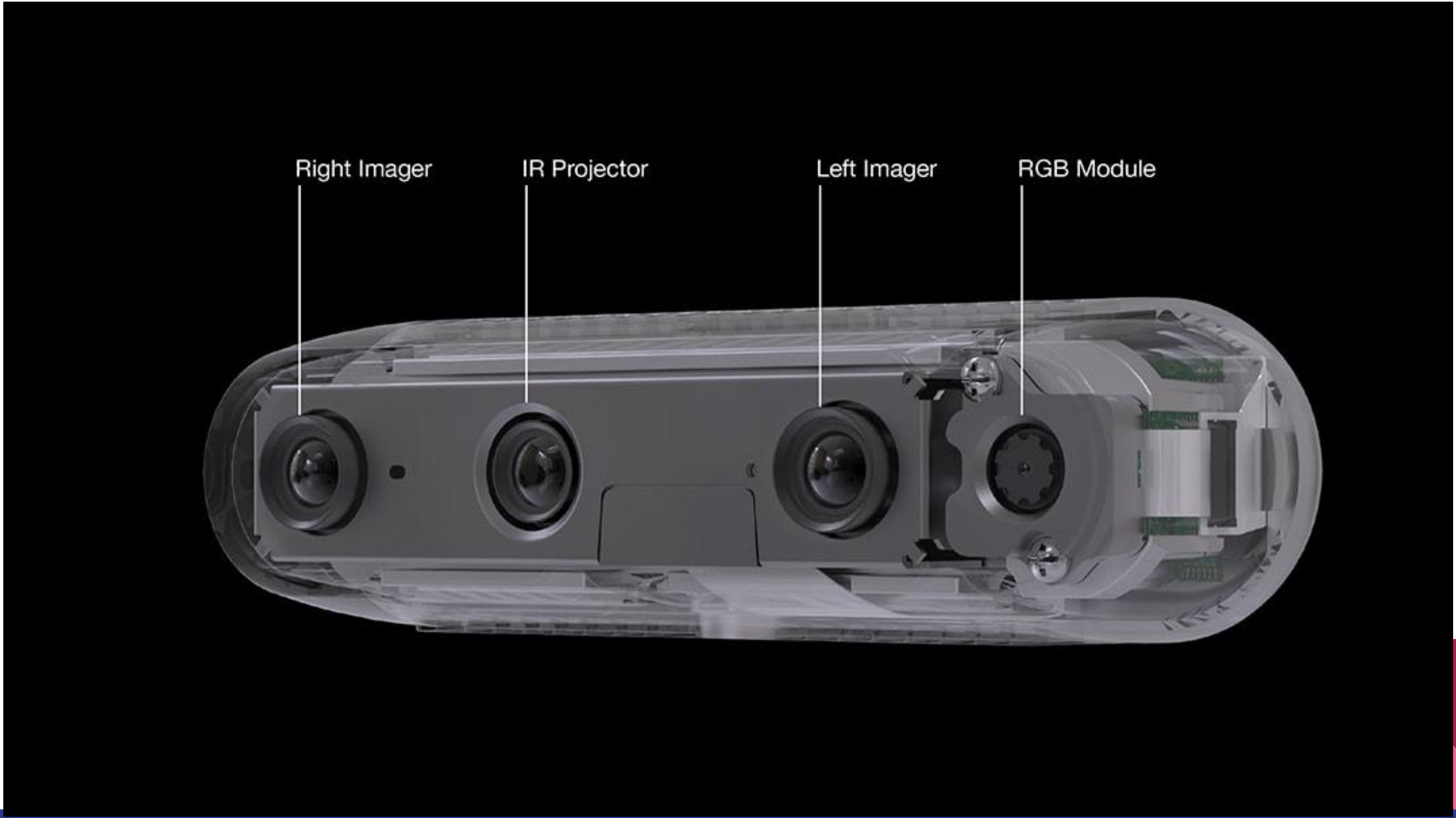
Given scores of 1-3 (higher is better)

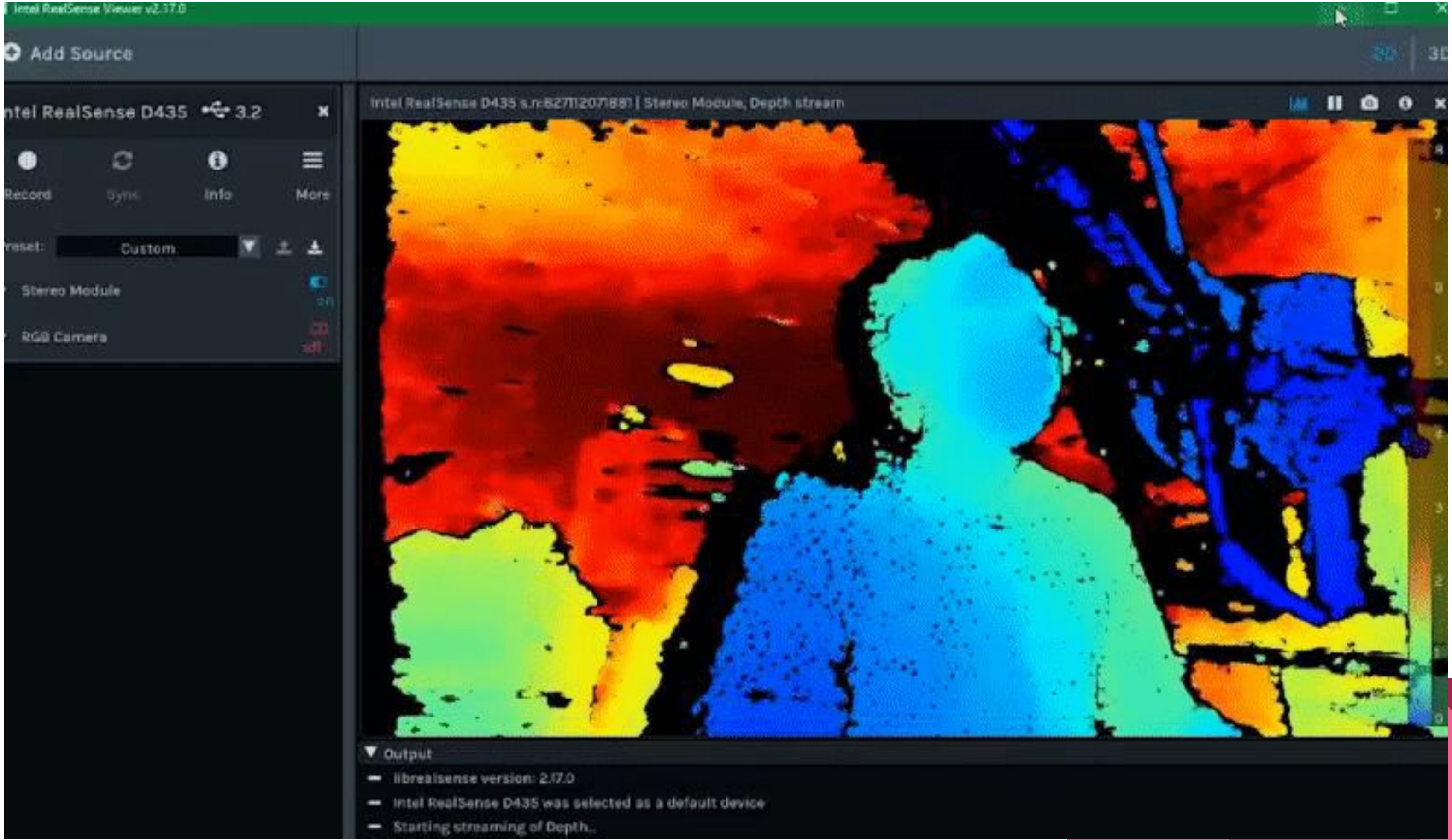
Weight	3	1	2	3
Type of Sensor	**Programming ease**	**Depth sensing**	**Image tracking**	**Affordability**
OpenMV Cam H7	3	1	3	2
Intel RealSense	**3**	**3**	**3**	**2**
Pixy 2	2	1	2	2
Weighted score	OpenMV Cam H7	Pixy camera	**Intel RealSense**	
	22	17	**24**	

Intel RealSense Depth Camera

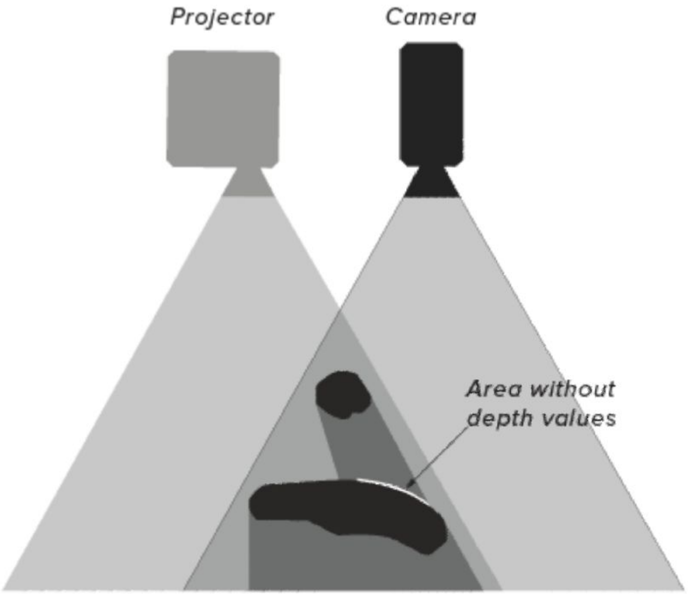
Intel RealSense Depth Camera




Right Imager
IR Projector
Left Imager
RGB Module

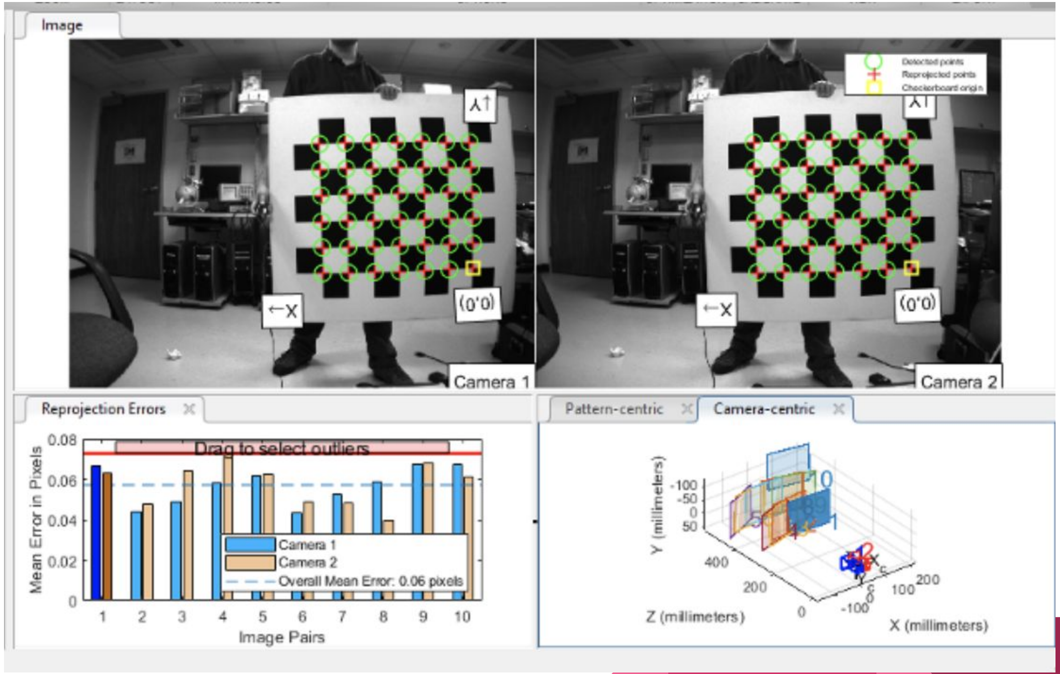

Intel RealSense Viewer v2.17.0
Add Source
ntel RealSense D435
3.2
Record
Sync
Info
More
Custom
Stereo Module
RGB Camera
Stereo Module, Depth stream
Output
librealsense version: 2.17.0
Intel RealSense D435 was selected as a default device
Starting streaming of Depth..

OCCLUSION SHADOWS



Computer Vision

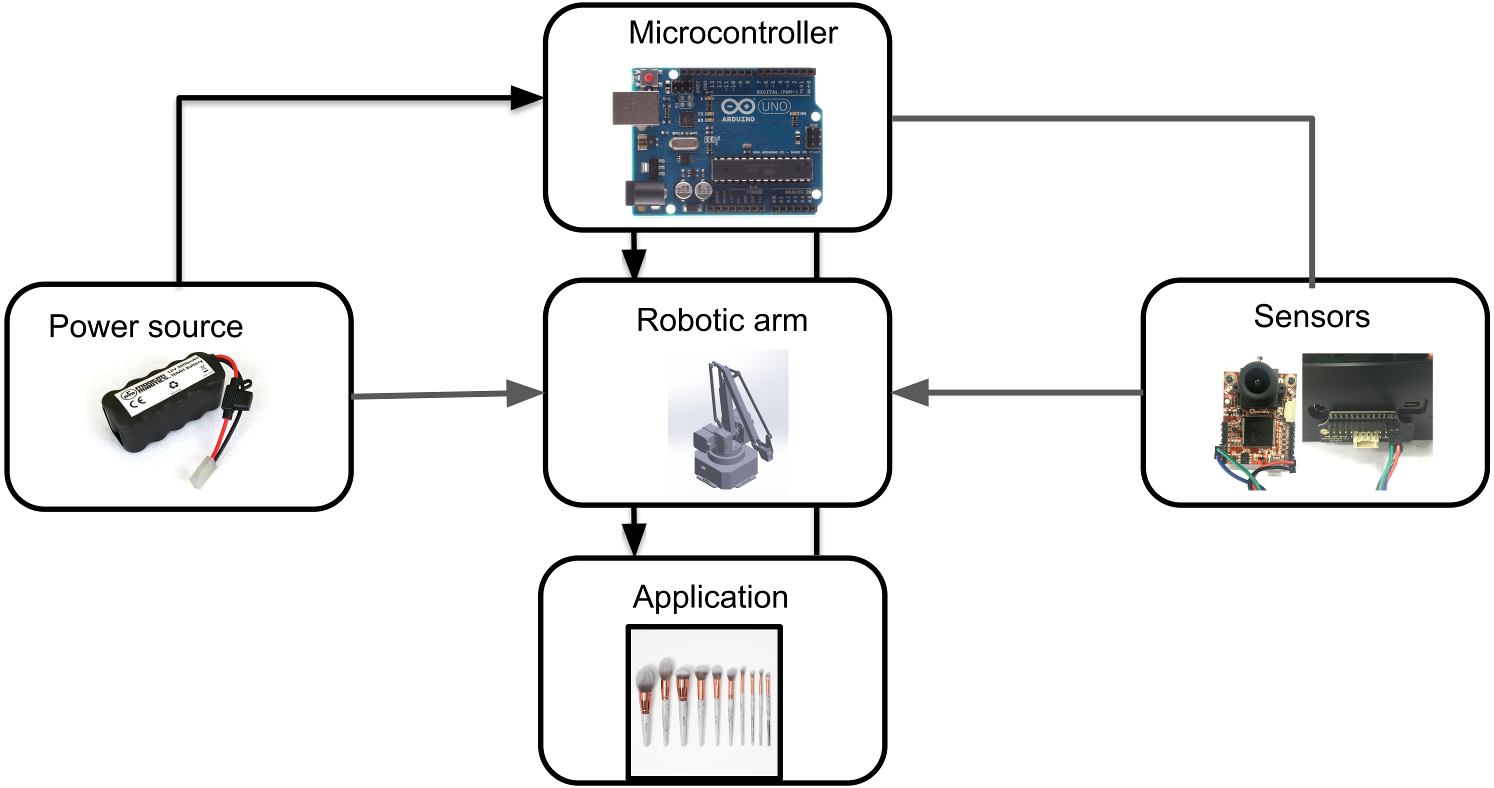
1. Active Stereo IR Camera
   a. Computes 3-D locations corresponding to matching pairs of image points using triangulation
   b. Crucial for depth features



Ultrasonic Sensor

- non-contact distance measurement module with a detection range of 3 cm to 4 m.
- Measure the distance between the robotic arm to the user.
- Plays a collaborative role.
- Compatible with the uArm Swift Pro.

System Schematic



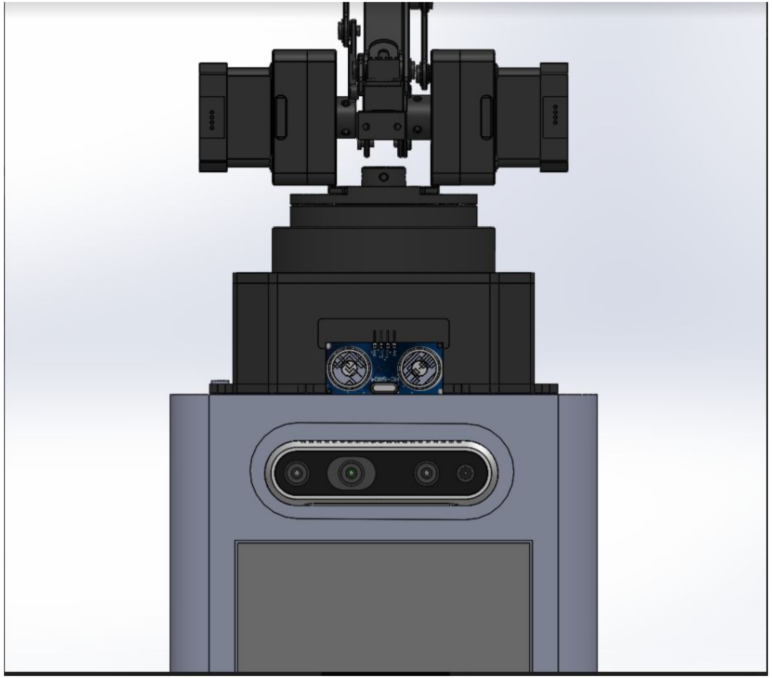
Modeling

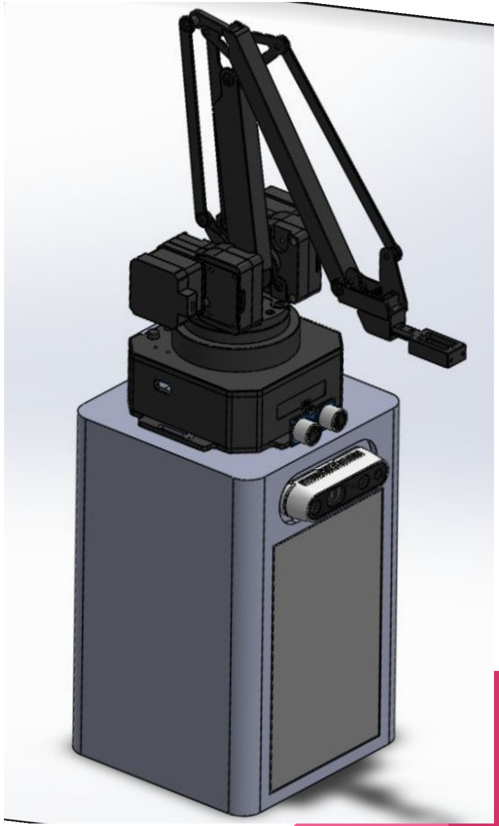
The CAD file for the uArm Swift Pro was obtained from the manufacturer.

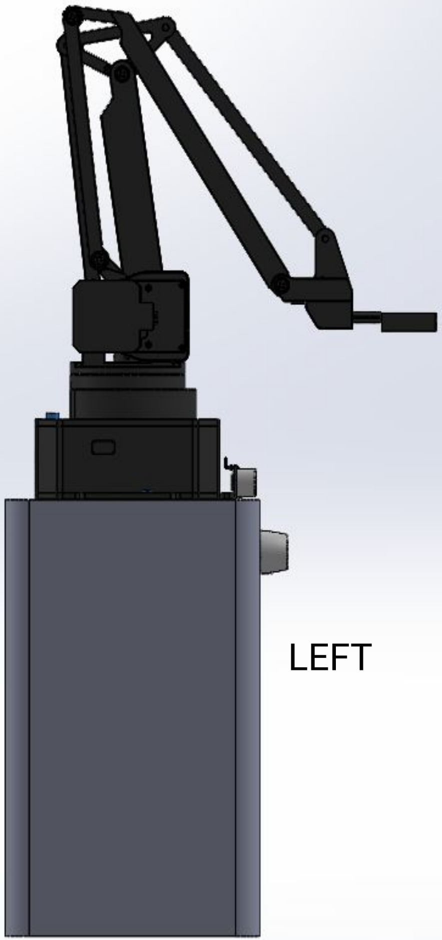
We plan to mount the robotic arm on a platform to raise its height, then secure sensors on openings in the platform.

The camera and sensor will be mounted at eye-level with the user sitting upright at ~1 ft away.

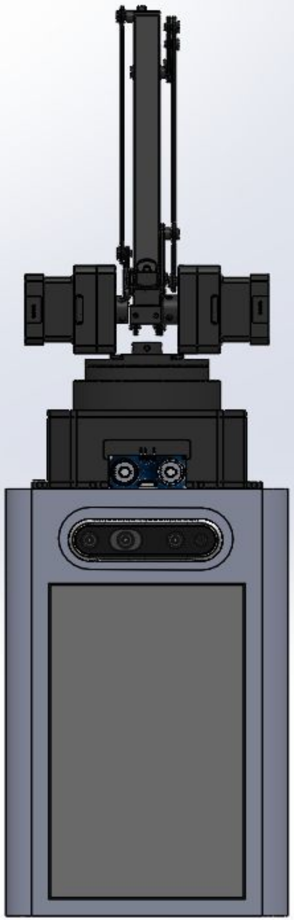
Modeling



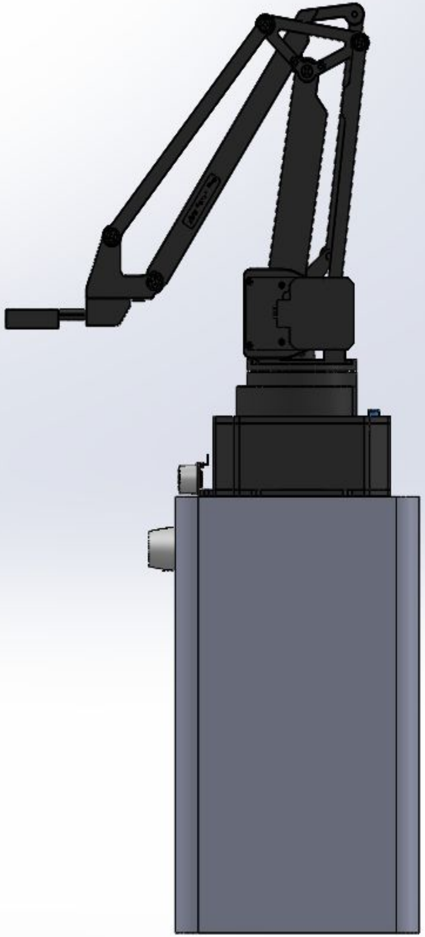




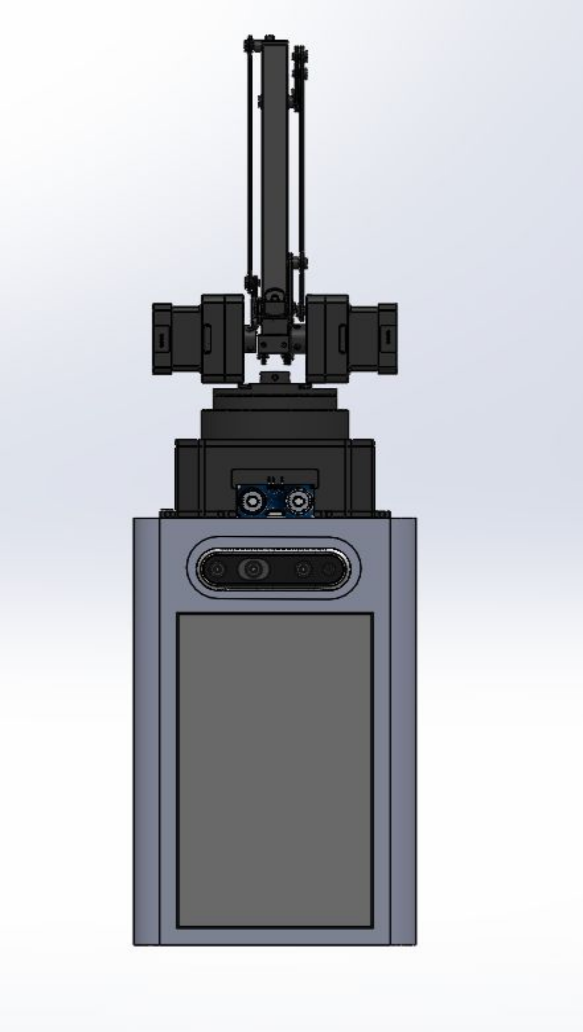
LEFT



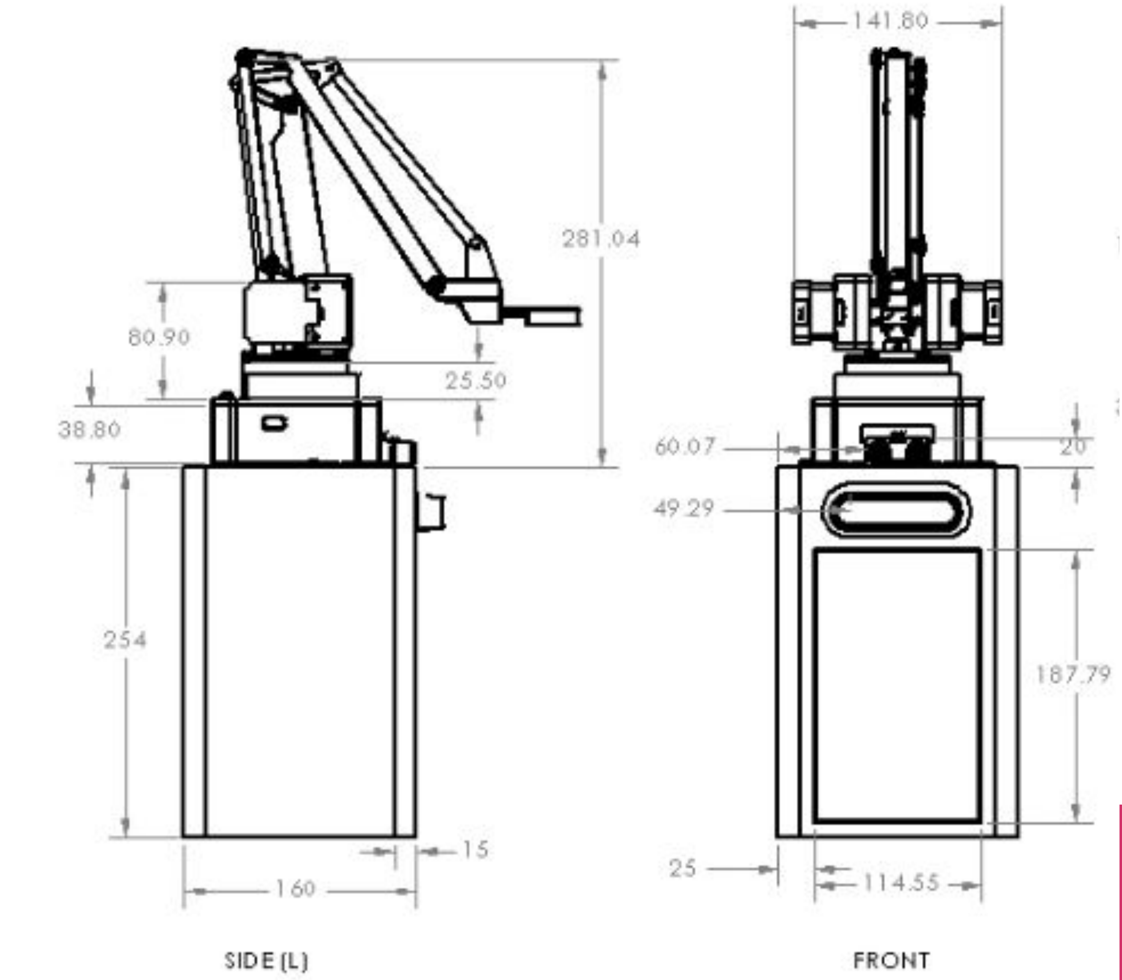
FRONT



RIGHT

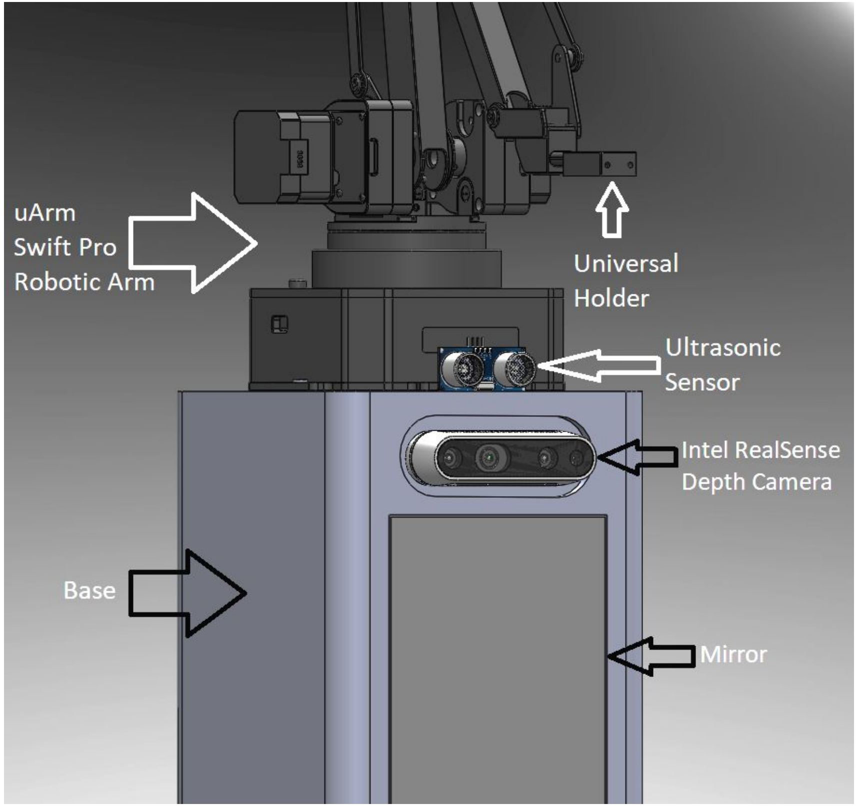


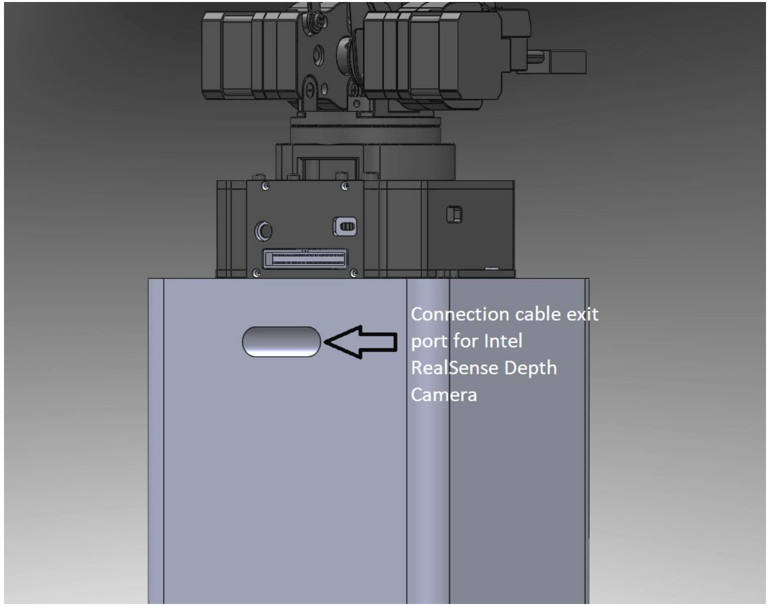
Modeling


141.80
281.04
80.90
25.50
38.80
60.07
20
49.29
254
187.79
15
160
25
114.55
SIDE (L)
FRONT

Modeling

Part notation



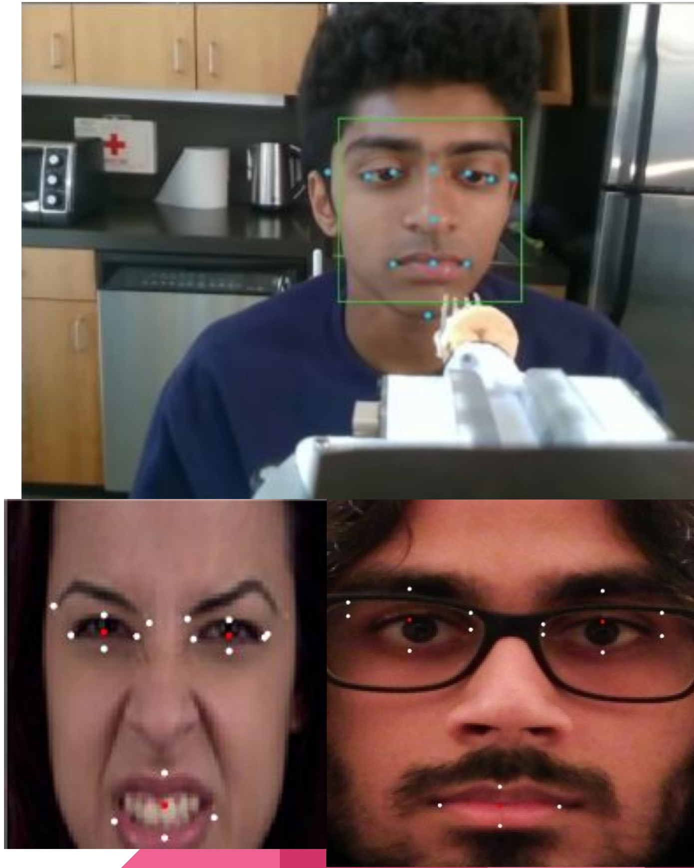


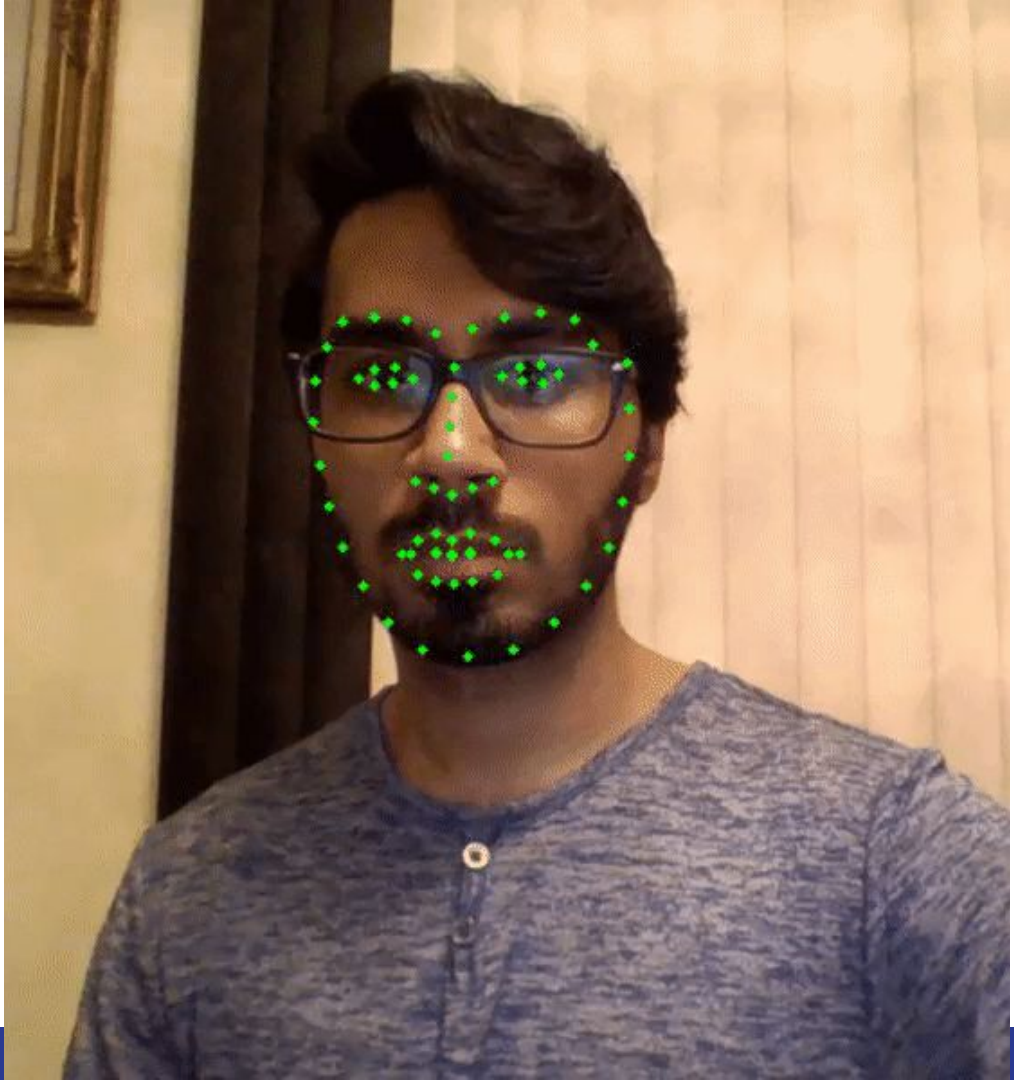
Landmark Tracking

The person’s face can be located using Facial Landmarks.

This can be done in real-time, as seen in examples of an autonomous feeding robot.

Currently we’ve been able to capture Landmark data for pictures, but will continue working on using this for live videos.





Our progress on Landmark Tracking

- track landmarks in real-time using Python and various external libraries
- Tracking starts when a person steps into frame, and stops when a person steps out a frame.
- Points stored as (x,y) coordinate
- Depth from Intel camera provides (x,y,z) point, and the camera can be programmed to move to points.

Further Steps

When setting the position of the uArm swift pro, it uses an arbitrary coordinate system which differs from the one derived by the sensors.

Since we don't yet have the robotic arm, we're not yet sure how easily it will be to translate our sensors' coordinates to the coordinates that can be interpreted by the uArm swift pro.

Once we get the arm, we'll be able to conduct tests on how the robot moves based on given inputs, and use that to determine how it should move based coordinates found from the sensors.

1. Budget
2. Manufacturing
3. Statistics
4. Testing


Q&A

References

Face statistics - https://www.facebase.org/facial_norms/summary/#palpfislength_r