ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

MassageMate: Smart Robot Masseur for ECE 445

Team #38

WENTAO YAO (wentaoy4)
KE XU (kex5)
XIUYUAN ZHOU (xiuyuan5)
HAO BAI (haob2)

TA: Yutao Zhuang

March 16, 2023

Contents

1 Introduction

1.1 Problem and Solution Overview

1.1.1 Problem

High-intensity work tends to cause fatigue in people's neck and waist, so they need massage to relax their shoulders and neck, but frequent visits to massage parlors cost quite time and are expensive. As an important component of Traditional Chinese Medicine (TCM), science of acupoint therapy benefits can include reducing stress and increasing relaxation, reducing pain and muscle soreness and tension, improving circulation, energy and alertness, lowering heart rate and blood pressure, and improving immune function. However, the number of acupuncturists who truly have qualified acupuncture skills is limited.

Besides, human massage also has some drawbacks and risks. Recognizing, positioning acupoints, and massaging is heavily depends on the skills of practitioners. Human massage may directly cause new injuries, mostly bruises and nerve lesions, aggravate existing injuries and chronic pain problems, distract patients from more appropriate care, mildly stress the nervous system, or even cause blood clot, nerve injury or bone fracture in rare cases. Some chronic pain patients may be disastrously traumatized by intense massage. Moreover, some customers may not want to be touched by an unfamiliar person, or they have some body privacy that hope to be hidden from others, then a Robotic Masseur [1] can help. Therefore, a machine or a robot that can perform the acupoint therapy automatically will reduce the workload of practitioners and benefit to people's daily health care.

1.1.2 Solution

The project utilizes Robotic arm and several sensors to massage users intelligently. We use a customized massage head, working together with the robotic arm to do massage. The user can control the robotic arm with voice instruction as well as some buttons through a Phone App. The NLP module will analyze user's input and to control the force, frequency as well as the position for the massage point. Moreover, the Acupoint detection model will detect the massage point based on traditional chinese medicine, and lead the arm move to that point. In addition, the vital signal sensor will also feedback the body condition information so that the robotic arm could adjust the massage force automatically.

For user interface application, the button in the upper-right corner is a switch, the center box shows the position of the masseur robot arm relative to the person, and the buttons in the lower-left corner are designed to move forward, downward, leftward and rightward, the "rotate" button can let a robotic arm knead, the swivel knob can reset the force, and the emergence key can pause the MassageMate.

The Robotic arm we use is OpenMANIPULATOR-P, controlled through DYNAMIXEL SDK through U2D2 connected to Raspberry Pi. We also designed the massage head, the head is basically a half-ball shape, fixed on a rack and pinion structure controlled by a high frequency motor. Another rack and pinion structure on the back of it will control the movement of massage head in vertical direction with the extension arm. The local control unit could adjust the massage force, frequency and height of this massage head, and receive sensors' outputs from it for processing.

We also plan to use a vital signs monitoring subsystem to detect whether the human feels comfortable or not. When the human feels uncomfortable, his blood pressure will go up. When this happens, we want the massage head to slow down its movement, thus making the person more comfortable.

Based on the Traditional Chinese medicine, there are several Acupoints (Acupuncture points) in human's body that is controlling human's feeling of pain as well as some symptoms, which is widely used in massage. We train a Computer vision model based on some open source pose models to detect human's Acupoints, and then let it to guide the Robotic arm to massage on certain locations.

1.2 Visual Aid

The figure 1 is the pictorial representation of our project

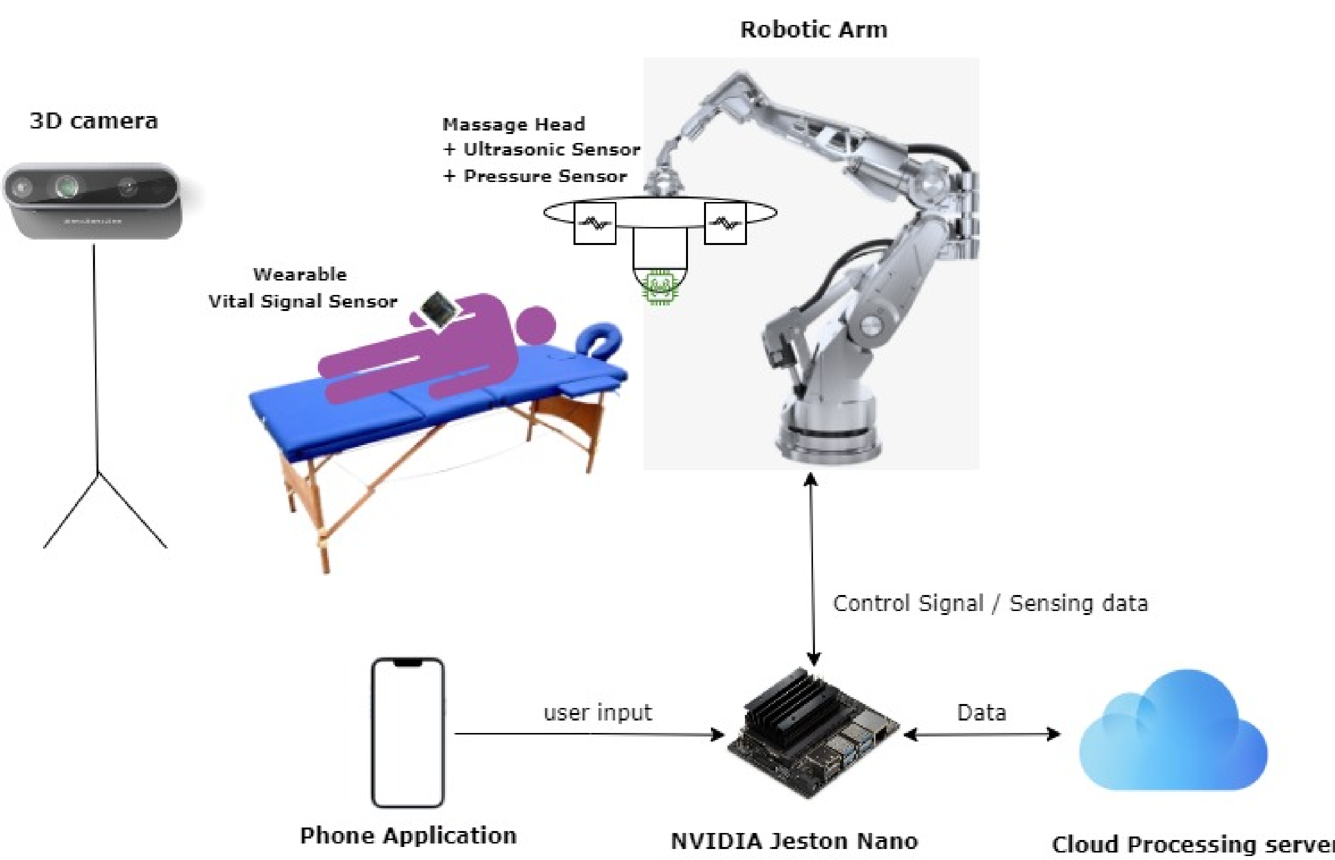

Figure 1: Visual Aid

1.3 High-level Requirements List

- The robotic arm and massage head can move according to the user's instructions and feedback, with appropriate distance, velocity, frequency and strength for massage. The ultrasonic sensors can adjust the position of the massage head when it detects a movement of the person being massaged, and a camera is applied to detect the Acupoint of the human.
- The phone app can record the user's voice and touch inputs and communicate with the cloud processing unit in real time, and the vital signs monitoring module can report whether the human is comfortable or not.
- The Processing unit can use ASR and finetuened Natural Language Models to understand the user's intention for massage, and generate massage instructions for controlling the robotic arm and massage head.

2 Design

2.1 Block Diagram

The figure 2 shows the block diagram of our system.

2.2 Description of Subsystem

2.2.1 Processing Unit

The processing unit in our system serve is the hardware that support collecting and processing data from sensors and robotic arm, running deep learning model inference, and control the robotic system. Therefore, the processing computer needs to contain GPU as well as having pin panels to receive signals from sensors.

We choose NVIDIA Jetson NANO 4GB3 as our processing unit. It has the layout as figure 4 shown. It is powered by 5V/3A power supply with Type-C power connecter. The jetson has many different types of hardware interfaces. The USB Type A will be used to connect the robotic arm, the 40-pin Exp Header is used to connect sensors in our system. 3D camera is connected to the Camera Conn Interface.

2.2.2 Robotic Arm Subsystem

The robotic arm we use in the project is OpenMANIPULATOR-P from ROBOTIS Co., Ltd.[2] This robotic arm is based on ROS and OpenSource, and it is composed of DYNAMIXEL-P modules. The DYNAMIXEL modules are the basic components of the robotic arm. They have a robotic modular form, and they can be combined together to form the arm with a chain method. To control the OpenMANIPULATOR-P from a PC, we use a U2D2 USB communication converter. This communication converter is fixed on a U2D2 Power Hub and connected with a TTL line. The Hub connects to the robotic arm also with a

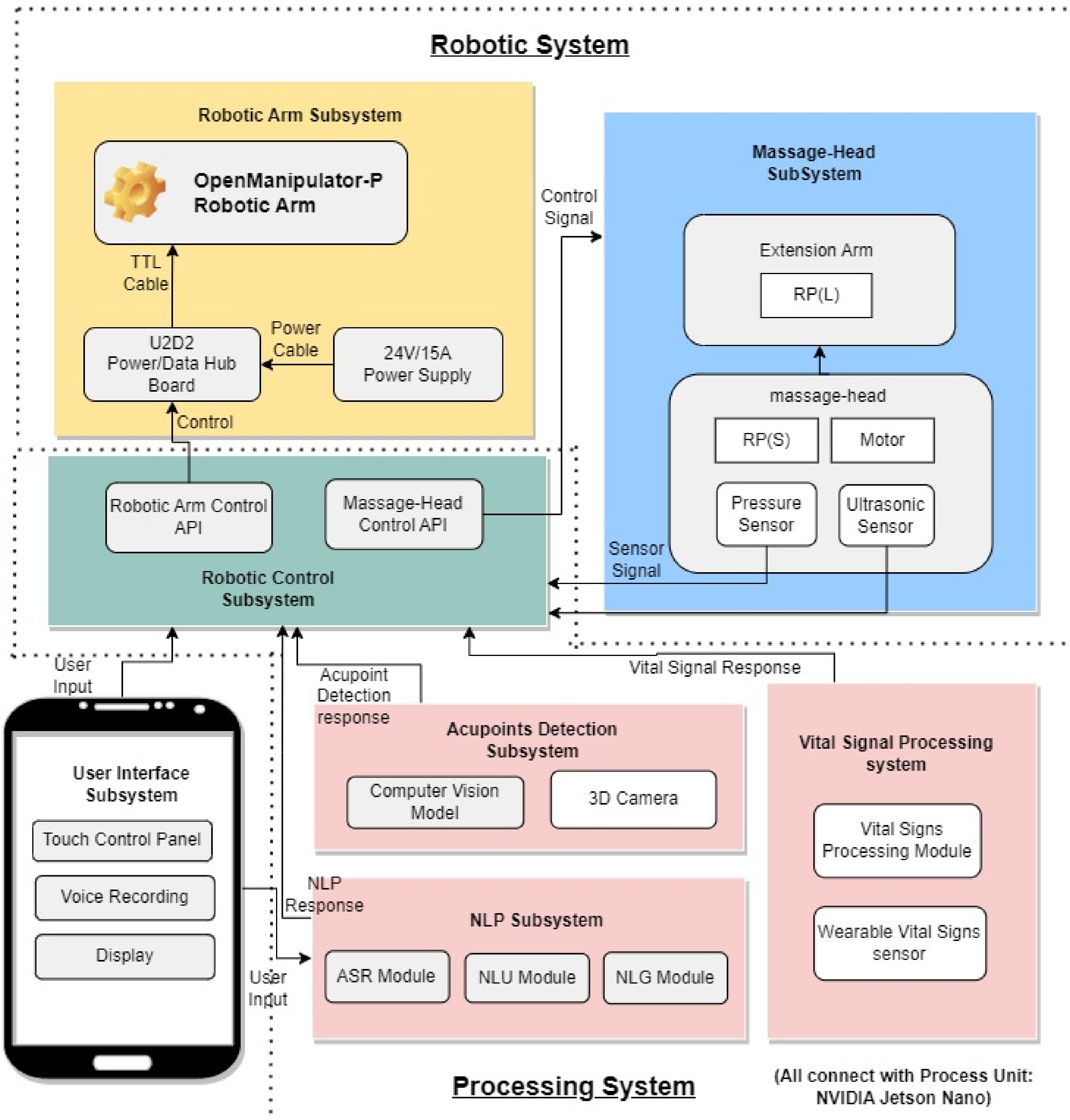

Figure 2: Block Diagram

TTL line, supplying power and sending control instructions. And the U2D2 will be connected to USB-A interface in Jetson board.The specification of robotic arm is shown in 5. It needs 24V/15A DC power supply, stationed at a movable base. This robotic arm has six-degree of freedom. It can generally reach nearly 700mm in length, with maximum

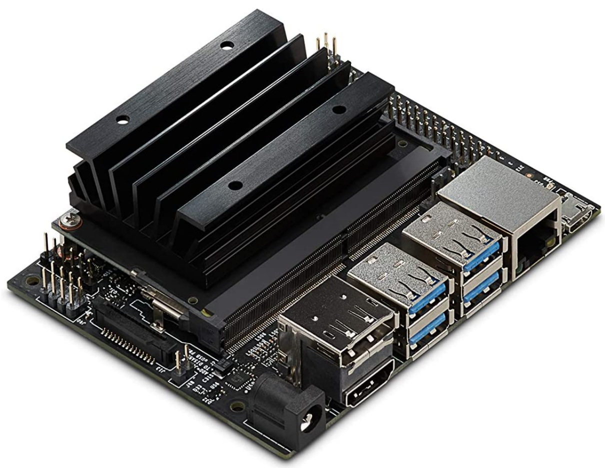

Figure 3: NVIDIA Jetson NANO

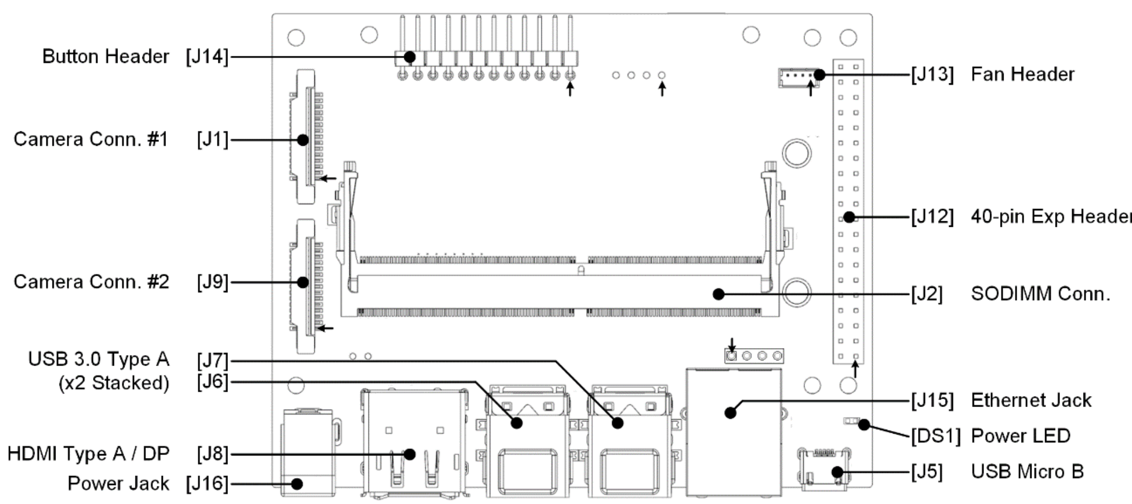

Figure 4: Jetson Layout

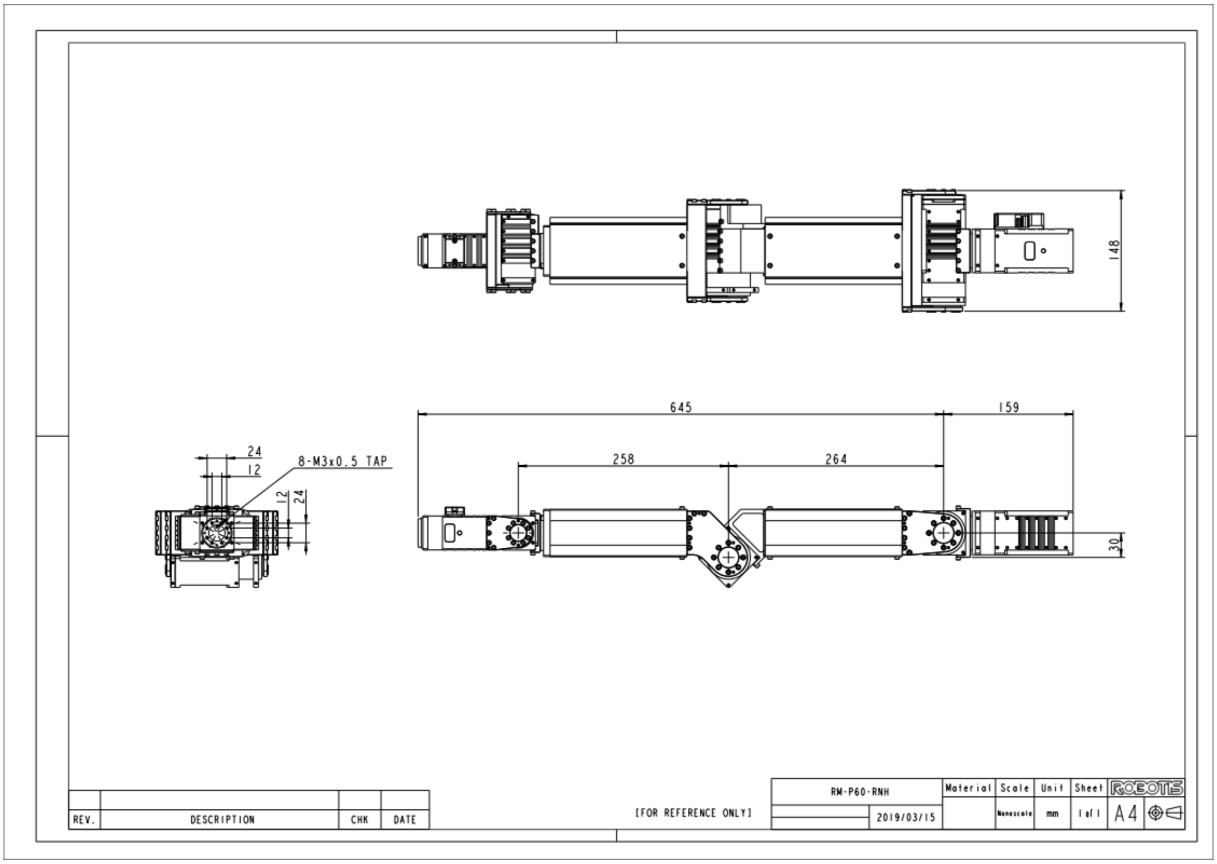

Figure 5: OpenMANIPULATOR-P Specification

payload 3kg.

2.2.3 Robotic Control Subsystem

The functionality for the control system is to gather and process all sensing data, and then generate massage policy to control the robotic arm and massage Head. It will connect directly to NLP subsystem, Acupoint detection subsystem, vital signal subsystem and sensors on massage head to receive input. Then, through the processing from the control program, it will send control instruction in robotic arm control API and massage head control API for controlling them separately.

a) Robotic Control

The robotic arm is connected with a communication converter called U2D2. It support ROS to control its motion in task space. It is connected as the figure 6 shows. To control the motion of robotic arm, we use the following ROS service:

- /open_manipulator/goal_task_space_path : Set the goal pose in task space at $pos = (x, y, z), Quat = (w, x, y, z)$
- /open_manipulator/goal_joint_space_path_from_present: Set the goal pose offset in

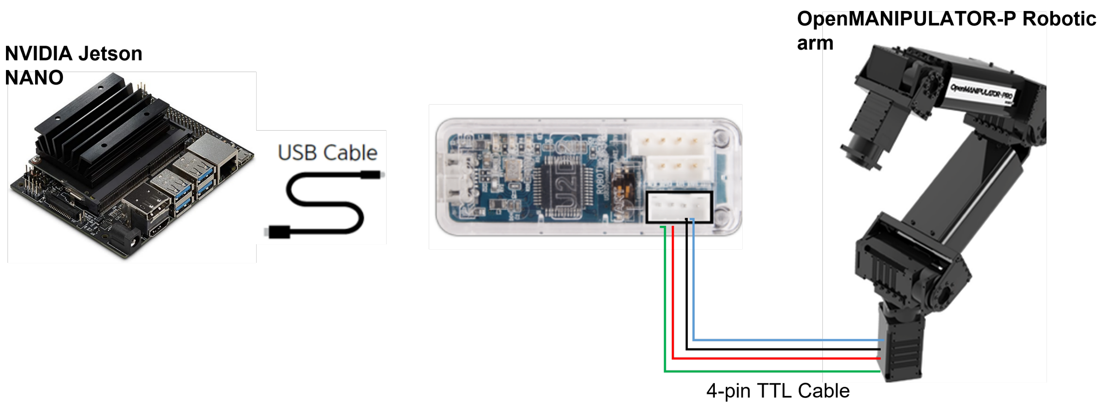

Figure 6: Robotic Arm connection

task space at $\Delta pos = (\Delta x, \Delta y, \Delta z), \Delta Quat = (\Delta w, \Delta x, \Delta y, \Delta z)$

And we also use the following ROS topics to obtain the status information of robotic arm in real time:

- /open_manipulator/joint_states : return the status of each joint $position : rad, velocity : rad/s$
- /open_manipulator/gripper/kinematics_pose: get the current pose information of robotic arm in task space: $Pos = (x, y, z), Quat = (w, x, y, z)$

We utilize those API for our control program of the robotic arm. Whenever the robotic arm receive the messages from perception subsystems and user inputs, it will then calculate the target location in both $pos = (x, y, z)$ and $Quat = (w, x, y, z)$ as well as the time for motion t to control the motion speed. And the control program will also get robotic arm pose information in real time, which could help the control program to adjust arm's motion.

b) Massage Head Control

The massage head is controlled by a single chip.

We set a standard pressure value and pressure sensor11 send back real-time message. Through a specialized chip, the message is translated to force value. Single chip compare the real-time force value with standard value. If real-time force value is larger than 110 percent of standard value or smaller than 90 percent of standard value, the robotic arm will move up or down the massage head until real-time force value is between 110 to 90 percent of standard value.

And single chip also controls the vibration of massage-head by control the high-frequency motor12.

Around the massage head is 2 ultrasonic ranging modules. When the measured distance value increase suddenly or two values has too much difference, they think message head leave body area, and force massage head stop moving.

2.2.4 Robotic Massage Head Subsystem

The massage head is consist of crankshaft construction and a rubber ball. Crankshaft includes eccentric gear7, rotate-rod8 and rectilinear moving rod9.

Offset 6mm of eccentric gear's center, rotate-rod connects with gear through a bearing (GB 608, OD 22mm and ID 8mm). the other end of rotate-rod connect with rectilinear moving rod through another bearing (GB 604, OD 12mm and ID 4mm). Rubber ball is connect to rectilinear moving rod with pressure sensor between them. 10

We use R370 motor12 to drive eccentric gear. When eccentric gear rotates, rotate-rod the goes up and down according to sine function, and push the rectilinear moving rod and rubber ball.

2.2.5 Acupoint Detection Subsystem

In recent years, researchers have proposed a few methods of automatic acupoints detection [3] [4] and positioning, but most of the methods are still based on manual designed features. With the advance of AI technology, we could do Acupoint Detection task with deep convolutional neural network.

The whole model structure is shown as 14. The original image of the human body and the tagged image are fed into the network. During training, the network generates a heat map of predictions. The loss between the predicted heat map and the real heat map is calculated based on the loss function of back propagation. When the model converges, the test image can be input into the network to obtain the predicted heat map. Finally, the point with the maximum response value was found on the heat map, and the coordinates were obtained as the location of the acupoint.

In this task, we will use Intel RealSense D435i Depth Camera shown as 15, which could offer quality depth for a variety of applications. With a range up to 10m, this small form factor camera can be integrated into any solution with ease, and comes complete with our Intel RealSense SDK 2.0 and cross-platform support. The general camera module setting and relavant parameters are shown as 16 and 17 respectively. And it will be connected to NVIDIA Jetson NANO with USB-C* 3.1 Gen 1.

2.2.6 NLP Subsystem

The NLP subsystem aims to use voice to control the massage head.

Modules

ASR (Automatic Speech Recognition)

This block receives signal input from the microphone from a phone or computer and converts the speech to text using the SpeechRecognition API with Python.

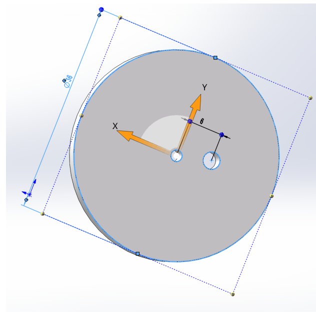

Figure 7: Eccentric Gear

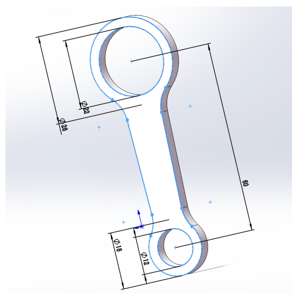

Figure 8: Rotate-rod

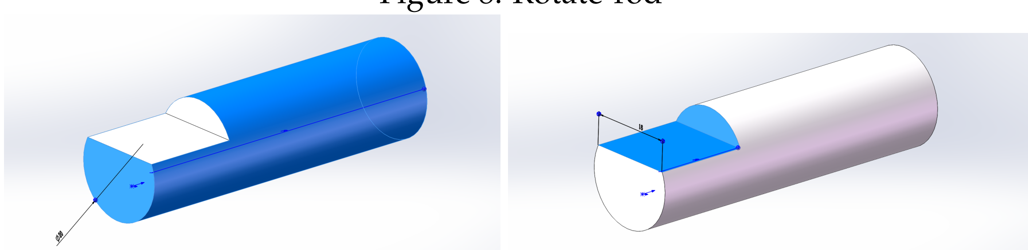

Figure 9: Rectilinear Moving Rod

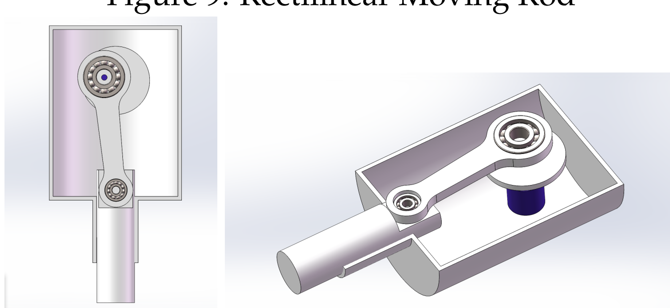

Figure 10: Massage Head

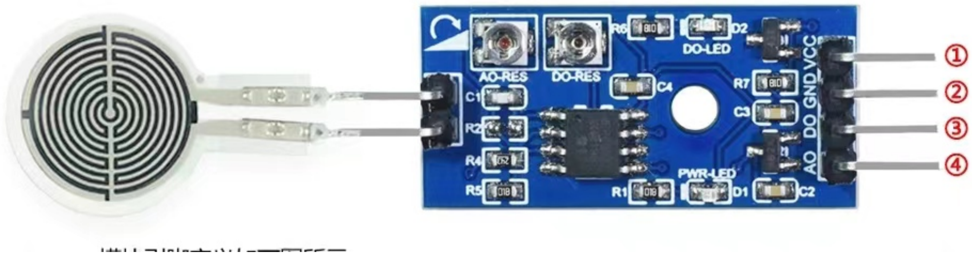

Figure 11: Pressure Sensor

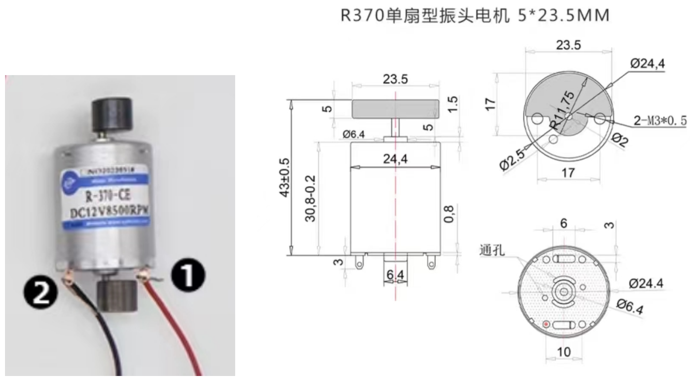

Figure 12: Motor

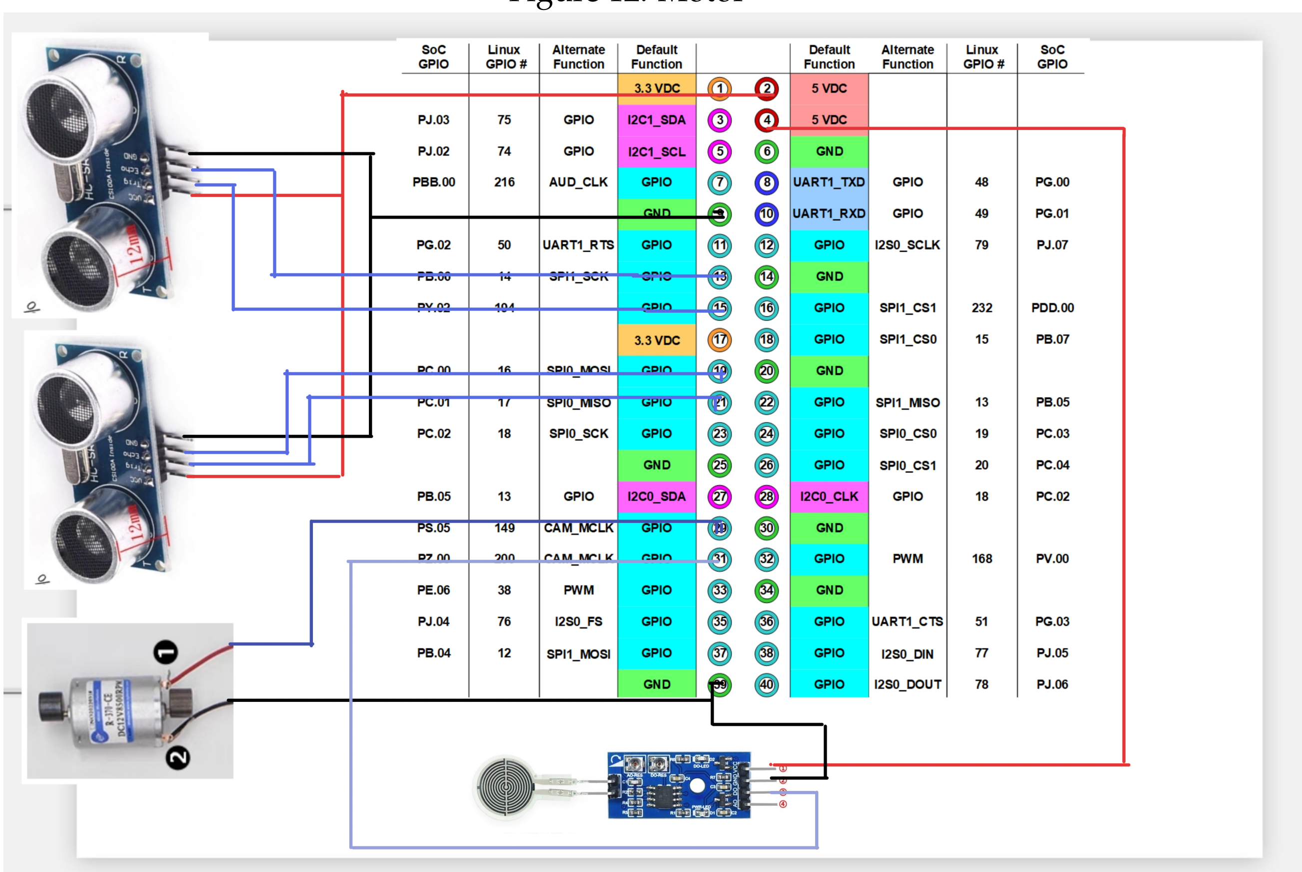

Figure 13: Massage Head Sensor Layout

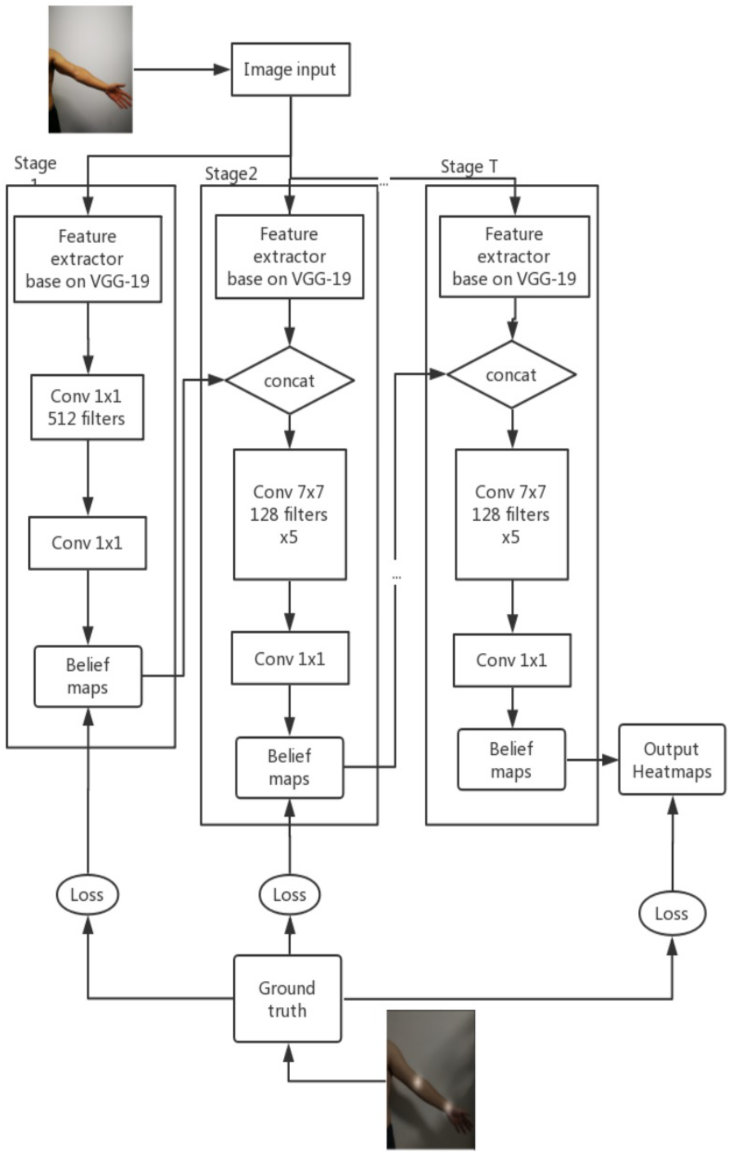

Figure 14: Convolutional Neural Network for Acupoint Detection

Figure 15: Intel RealSense D435i Depth Camera

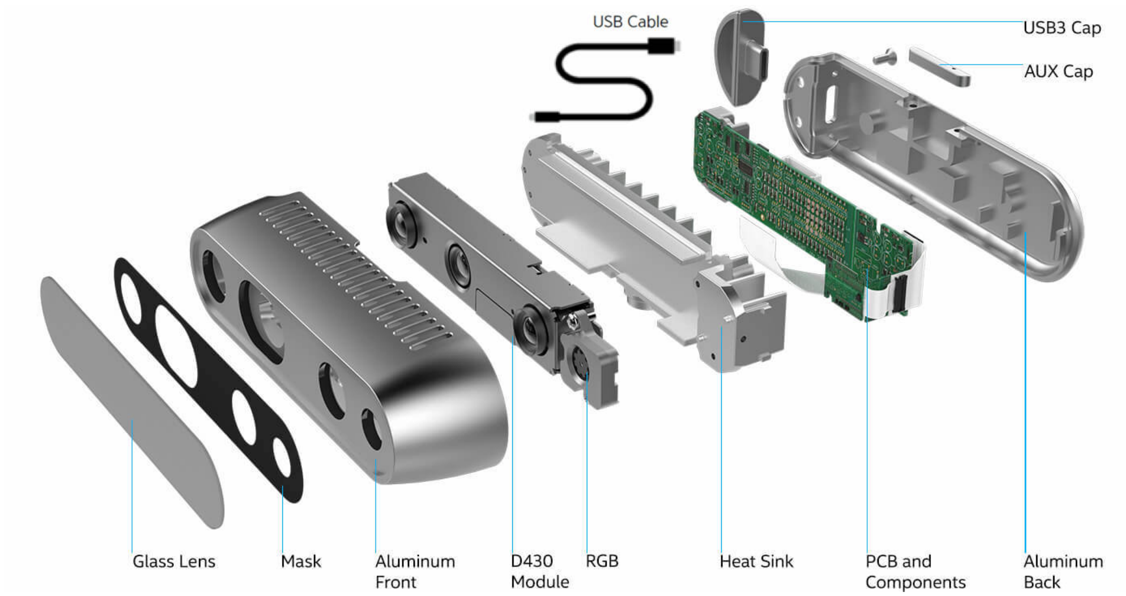

Figure 16: Intel RealSense D435i Depth Camera Structures

STT (Speech-To-Text)

This block receives the text response from the system and makes a sound through the speaker on the phone or computer, with the pytts3 library using Python.

NLU (Natural Language Understanding)

Block Description: The NLU module is responsible for understanding the user's natural language input, such as speech or text. It performs several tasks such as tokenization, named entity recognition, intent classification, and slot filling [5].

Contribution to Overall Design: The NLU module is critical to the overall design of the task-oriented dialogue system as it is the initial step in processing the user's input. It extracts important information such as the user's intent and any relevant entities, which is then used by the subsequent modules to generate an appropriate response.

Interfaces with Other Blocks: The NLU module interfaces with the DST module to provide information about the user's intent and any relevant entities. It also interfaces with the PM module to provide information about the user's input.

DST (Dialogue State Tracking)

Block Description: The DST module keeps track of the current state of the conversation. It uses the output from the NLU module to update the dialogue state, which represents the current context of the conversation.

Contribution to Overall Design: The DST module is critical to the overall design of the task-oriented dialogue system as it maintains the current state of the conversation. It keeps track of the user's goals, preferences, and any relevant information that is needed to generate an appropriate response.

Features	**Use environment:** Indoor/Outdoor	**Ideal range:** .3 m to 3 m
Depth	**Depth technology:** Stereoscopic **Minimum depth distance (Min-Z) at max resolution:** ~28 cm **Depth Accuracy:** <2% at 2 m[1] **Image sensor technology:** Global Shutter	**Depth Field of View (FOV):** 87° × 58° **Depth output resolution:** Up to 1280 × 720 **Depth frame rate:** Up to 90 fps
RGB	**RGB frame resolution:** 1920 × 1080 **RGB frame rate:** 30 fps **RGB sensor technology:** Rolling Shutter	**RGB sensor FOV (H × V):** 69° × 42° **RGB sensor resolution:** 2 MP
Major Components	**Camera module:** Intel RealSense Module D430 + RGB Camera	**Vision processor board:** Intel RealSense Vision Processor D4
Physical	**Form factor:** Camera Peripheral **Length × Depth × Height:** 90 mm × 25 mm × 25 mm	**Connectors:** USB-C* 3.1 Gen 1* **Mounting mechanism:** – One 1/4-20 UNC thread mounting point. – Two M3 thread mounting points.

Figure 17: Intel RealSense D435i Depth Camera Technical Parameters

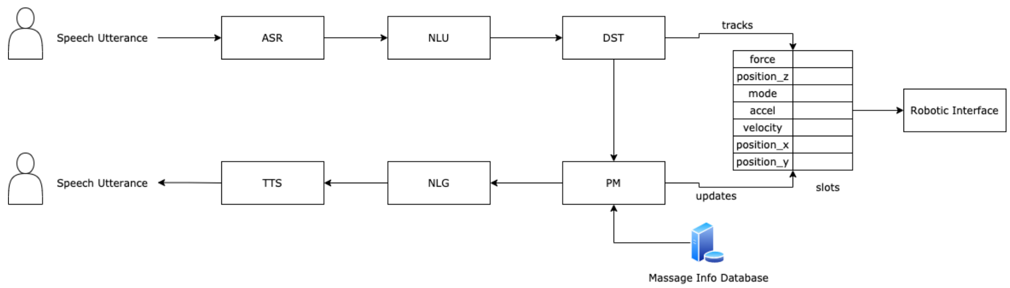

Figure 18: The overall architecture of the components of the NLP subsystem.

Interfaces with Other Blocks: The DST module interfaces with the NLU module to receive information about the user's intent and any relevant entities. It also interfaces with the PM module to provide information about the current state of the conversation.

PM (Policy Manager)

Block Description: The PM module is responsible for determining the appropriate response to the user's input based on the current dialogue state. It uses a combination of rules and machine learning algorithms to generate a response that meets the user's goal and takes into account the current context of the conversation.

Contribution to Overall Design: The PM module is critical to the overall design of the task-oriented dialogue system as it generates the response to the user's input. It takes into account the user's goal, preferences, and the current context of the conversation to generate an appropriate response.

Interfaces with Other Blocks: The PM module interfaces with the DST module to receive information about the current state of the conversation. It also interfaces with the NLG module to provide information about the response to be generated.

NLG (Natural Language Generation)

Block Description: The NLG module is responsible for generating the final response to the user's input in natural language. It takes the output from the PM module and generates a response that is grammatically correct, semantically meaningful, and appropriate for the context of the conversation.

Contribution to Overall Design: The NLG module is critical to the overall design of the task-oriented dialogue system as it is responsible for generating the final response to be presented to the user. It takes into account the information provided by the PM module and generates a response that is clear, concise, and appropriate for the context of the conversation.

Interfaces with Other Blocks: The NLG module interfaces with the PM module to receive information about the response to be generated. It also interfaces with the user interface module to present the final response to the user. [6]

Train

`DST` does not need to be trained - it's simply a table in our task.

`NLU` and `PM` will construct one model. We currently plan to use `ChatGPT` to distill a `Flan-T5` model on the given slots. The training input should be `(user command, current state)` and the output should be `(update of current state)`. The training data will be generated by `ChatGPT` and we will manually finetune the `Flan-T5` model on the synthetic data. Note that we will control the change of `DST` to ensure that the change is acceptable and safe for the person being massaged.

`NLG` will construct another model. The input is `(user command, current state, update of current state)`, and the output should be `(bot feedback utterance)`. This model can be directly distilled from `ChatGPT` and only slight manual work is required to ensure the safety.

We will NOT use ChatGPT directly for this task because of the RISKY zero-shot performance. We want to make sure the human being massaged is safe.

Inference

An inference example is shown below. The arm controls the blue slots and the dropper controls the green slots.

`force`: massage force.

`position_z`: massage head (dropper) height.

`mode`: can be `OFF` (off), `LR` (left-right), `ROT` (rotation).

`accel`: acceleration and smoothment.

`velocity`: maximum horizontal velocity.

`position_x`: massage head x position.

`position_y`: massage head y position.

Given a user utterance, the first model updates the slots, while the second model call the TSS model for speech output.

2.2.7 Vital Signal Processing system

The vital signal processing system uses a watch wearing on the human's wrist. The watch is composed of one chip (Vital Signs Monitoring Chip) MQ-R-12. The chip is shown below.

The chip layout is shown below.

After gathering data, we plan to collect data including the person's heart rate, blood pressure, oxygen saturation, and temperature. The MQ-R-12 chip includes various sensors to detect these vital signs, including an electrocardiogram (ECG) sensor for heart rate monitoring, a photoplethysmogram (PPG) sensor for blood pressure and oxygen saturation monitoring, and a thermopile sensor for temperature monitoring.

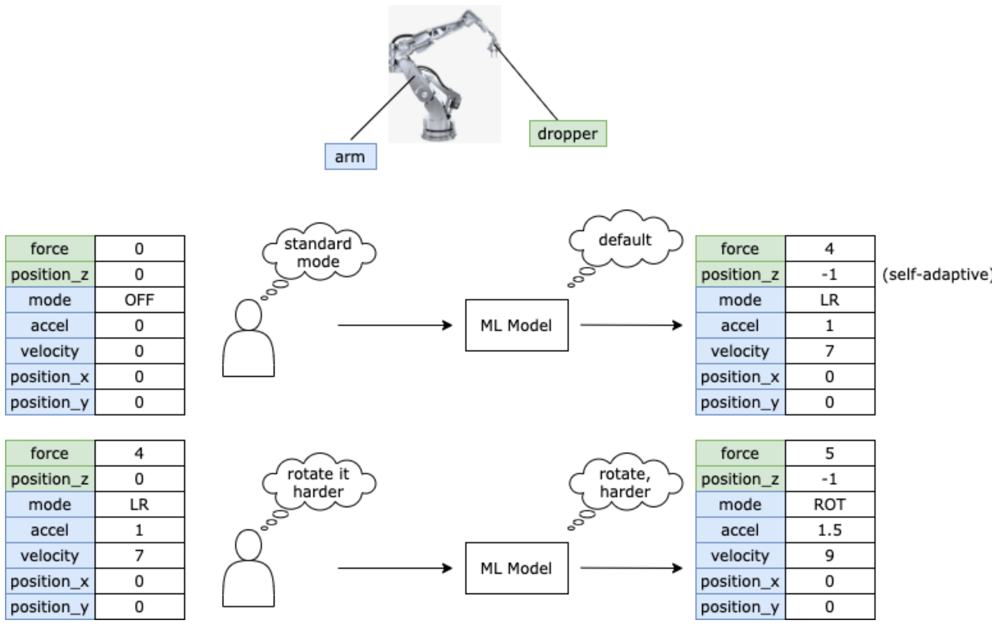

Figure 19: The illustration of the overall architecture of the working pipeline.

Once the data is collected, it is processed using algorithms and software to extract meaningful information about whether the person wants to make the force larger or not, etc. This is done through collecting data from real-life massaging, while the massaged person wears this chip.

When the model detects a bad feeling, the robotic arm will adjust its parameters to better suit the person, so as to achieve a **closed-loop** control.

2.2.8 User Interface Subsystem

As shown in 22, the button in the upper-right corner is a switch, the center box shows the position of the masseur robot arm relative to the person, and the buttons in the lower-left corner are designed to move forward, downward, leftward and rightward, the "rotate" button can let a robotic arm knead, the swivel knob can reset the force, and the emergence key can pause the MassageMate.

2.3 Requirements and Verification

2.3.1 Processing Unit (Jetson Nano)

- Requirements
  1. The Jetson Nano has enough Pins and interfaces to connect sensors and the 3D camera. The connection on board should be stable under the motion of robotic arm and massage head.
  2. Jetson Nano could have enough CUDA memory to support the inference of

Figure 20: The Vital Sighs Monitoring Chip MQ-R-12 pictorial graph.

序号	信号名称	功能	说明
1	VCC_IO	数字 IO 供电	2.7V ~ 3.6V
2	RST	复位	低电平有效
3	URX	模块 UART 接收	电平与 VDD_IO 一致
4	UTX	模块 UART 发送	电平与 VDD_IO 一致
5	STA	保留	悬空
6	GND	GND	

Figure 21: The Vital Sighs Monitoring Layout.

Acupoint Detection model and Vital Signs Processing model simultaneously. And the control program could also run at the same time to process data and generate massage policy.

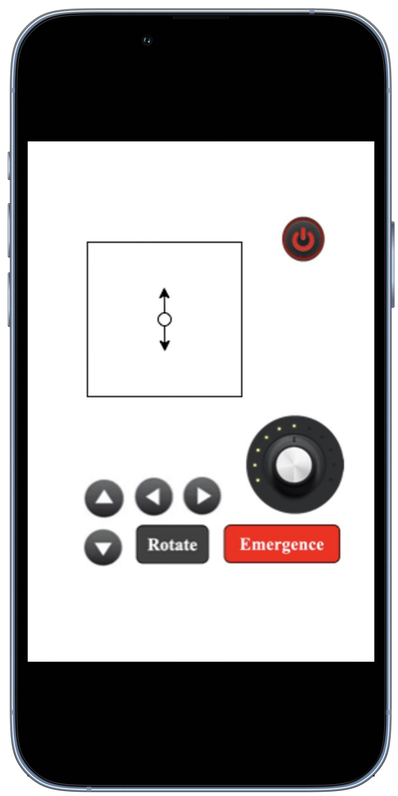

Figure 22: Phone App for Users

- Verification
  1. Connect all sensors/camera onto the Jetson Nano board, running the motion of robotic arm and massage head randomly, checking if obtaining data and sending control signals could work properly.
  2. Running the Control program and machine learning models at the same time, and doing some functionality tests to check if the Processing Unit could work fine.

2.3.2 Robotic Control & Robotic Arm Subsystem

- Requirements
  1. The control subsystem could receive data from sensors and cameras as well as user's voice input correctly in real-time. The delay for receiving sensor data should below 20ms, and for receiving user's voice input, the delay should below 100ms.
  2. The control of robotic arm and massage head work properly in processing all input data and generate appropriate massage policy, which set the parameter for the motion of both massage head and robotic arm. The control program processing time should below 200ms. The control signal delay between Jetson Nano and Robotic arm should below 10ms. To ensure safety, in emergency case (when user want to stop immediately), the latency of control program processing plus control signal sending should below 100ms.
- Verification
  1. For safety concern, we firstly run our test cases of robotic control in software simulation. This includes some basic control program test like to judge if robotic arm could move correctly under the voice instruction or Acupoint detection. Then, we move the test in real cases on a soft mannequin. In that stage, we add the tests on the feedback control, including pressure sensors and distance sensors to make sure the massage force on human could be appropriate. Finally we will do some tests on human body to make sure the functionality could achieve as well as guarantee the safety.

2.3.3 Voice Controller Subsystem

- Requirements
  1. We aim to achieve a safe and accurate control using voice, and provide the user with helpful feedbacks.
- Verification
  1. To test whether we have successfully achieved our goal, we will invite several students to do human evaluation on whether the system is robust and helpful.

2.3.4 Vital Signs Monitoring

- Requirements
  1. The final system should be able to adjust the robotic arm based on the statistics it collect from the vital chip.
- Verification
  1. The average slot filling accuracy (ASFA) should be larger than 90 percent from the dataset we collect.

2.3.5 Massage Head Subsystems

- Requirements
  1. Pressure sensor must fast enough to send back real-time data.
  2. Elements must have enough mechanical strength so that when the massage head vibrates, it will not be broken.
  3. Motor must have high enough frequency and low noise.
  4. When head leave body area, stop at once.
- Verification
  1. When tested on a dummy, the pressure stabilizes in the vicinity of the standard value.
  2. When dummy is moved suddenly, stop at once.
  3. It can work at maximum frequency for a period of time without the structure falling apart.
  4. When tested on human body, massage makes user feel comfortable.

2.3.6 Acupoint Detection Subsystem

- Requirements
  1. The distribution of acupoints in the human body is highly dependent on the key points of the human skeleton. Therefore, our model should perform well in leaning local and spatial features.
  2. A relatively large dataset of human body images is vital. What's more, an evaluation index for acupoint detection is also very important when the experiments are conducted.
- Verification
  1. The code should be executed without bugs so that model can be trained and predictions can be made.

响应时间	< 1ms
恢复时间	< 15ms

Figure 23: sensor

2. The model should perform well so that acupoint can be located accurately.

2.3.7 User Interface Subsystem

- Requirements
  1. The user interface should be simple and easy to understand considering easy access and operation.
  2. Overall response time should be rapid and the delay of information transmission to the server should be small.
  3. Security must be a high priority to prevent users from receiving fraud and abuse that endangers hardware, data, and even human security.
- Verification
  1. Do one thousand click test, can carry out smooth signal transmission, time error within 100ms.

2.4 Tolerance Analysis

In order to achieve the functionality of controlling robotic masseur, the system have following data for the analysis:

volume of metal part:
eccentric gear:$7035mm^3$
rotate-rod:$5485mm^3$
rectilinear moving rod:$20468mm^3$
total mass of metal part is about $0.00785(7035 + 5485 + 20468) = 259g$
the remaining part is some small sensor.
The head of the arm can mostly lift up to 30 kilograms, and mass of massage head with two ultrasonic ranging modules is no more than 1 kilogram, so the robotic arm can list massage head easily.

response time of sensor is less than 1 ms
response time of robotic arm's control module is about 16 ms
total response time is less than 20 ms, is short enough to adjust position and force to comfort user.

For the Acupoint detection system, since the camera is setup 1 meter above human, the model need to be able to detect keypoint with the rescale of image. The model accuracy need to reach above at least 60%, and the frame rate should be above at least 45FPS.

The Vital signal module should filter the noise from detection, there should be at least 5 clean features for model to inference. And the average slot filling accuracy could reach at least 90 percent in the evaluation set for those 5 features.

3 Cost and Schedule

3.1 Cost Analysis

Our fixed development costs are estimated to be ¥40/hour, 10 hours/week, in total 12 weeks for four people. The total human cost is ¥19200.

The main parts are concentrated in robotic arm. Our costs are estimated as follow:

Name	Quantity	Unit Price	Total Cost
NVIDIA Jetson NANO 4GB	1	¥1299	¥1299
OpenMANIPULATOR-P Robotic Arm	1	¥96766	¥96766
Intel RealSense D435i 3D Camera	1	¥2829	¥2829
Ultrasonic ranging module	2	¥4.8	¥9.6
RP-C18.3-LT Pressure sensor	1	¥29.8	¥29.8
Pressure sensor data translation module	1	¥8	¥8
R370 motor	1	¥8.5	¥8.5
Metal part in message head	1	about ¥400	about ¥400

As for NVIDIA Jetson NANO, robotic arm and 3D camera, we can use components that are readily available in the school laboratory.

Ultrasonic ranging module, pressure sensor and motor are purchased from companies, and may not reach the requirements for our design. Each part listed may be broken when they are transported or used. So we need flexible budget to choose and purchase new elements.

The total cost is about ¥120549.9.

3.2 Schedule

3.2.1 Overall Schedule

- By the end of March

  - Purchase all sensors/devices/control board for the project. Build up and connect all components.
  - Explore & train the Acupoint Detection model, integrate it into the system.
  - Finish the first version of NLP model. Test its ability in understanding human's instructions.
  - Finalize the design of massage head, manufacturing parts of it.

- To mid April
  - build all "smart part" of the system (NLP submodule, acupoint detection, vital signal detection model)
  - Finish the prototype of smart robotic masseur. Doing safety/functionality tests on it.

3.2.2 Per-person

Week Number	Wentao	Ke	Xiuyuan	Jack
3.19-3.26	Inital set up of robotic arm	Inital set up of robotic arm	study how to transmit singles to control systerm	Survey slot filling and dialog system.
3.27-4.2	Write Python API for the control of robotic arm	Study acupoint detection method	Make metal part of massage and connect it to robotic arm	Survey vital signs monitoring subsystem.
4.3-4.9	Connect and test all sensors	Connect Camera and try different detection methods	Make metal part of massage and connect it to robotic arm. Able to control its motor	Implement slot filling and dialog system.
4.10-4.16	Finish the feedback control for sensors on massage head	Evaluate different methods and choose the best one to optimize	Study relationship between strength and position of massage and user's confort level	Implement vital signs monitoring subsystem.
4.17-4.23	Integrate robotic control with acupoint detection, doing camera calibration	code user-interface for control through phone App or handle	Improve pressure control	Integrate the slot filling and dialog system.
4.24-4.30	Integrate the Vital signs subsystem with robotic control	Do real-human safety and functionality test	Test mechanical reliability test	Integrate vital signs monitoring subsystem.
5.1-5.7	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo
5.8-5.22	Improve the functionality and write final report	Improve the functionality and write final report	Improve the functionality and write final report	Improve the functionality and write final report

4 Discussion of Ethics and Safety

4.1 Ethics

Developing a project that involves the use of robotic arms for massaging humans presents both ethical and safety considerations. The ethical issues that need to be considered include issues of privacy, informed consent, and the potential for harm to the client.

The IEEE Code [7] of Ethics emphasizes the importance of the safety and welfare of the public. Therefore, the project should prioritize the safety and comfort of the human clients. The ACM Code of Ethics [8] emphasizes the importance of avoiding harm, and the project should strive to avoid any potential harm to the clients.

To avoid ethical breaches, the project team should obtain informed consent from the clients before beginning the massage. The clients should be informed of the benefits and risks of the massage, as well as any potential discomfort or pain that may arise. The project team should also respect the privacy of the clients by ensuring that any personal information obtained during the massage is kept confidential.

We also need to keep the privacy of the person being massaged. In our vital signs controller and voice controller module, we will develop datasets which concerns privacy. Thus, we rule that all collected data should not include anything about the identity of the human being experimented (e.g. name, ID, etc.).

We also notice that the figure of the human may affect the design of the massage head. Thus, we will try to make the massage head able to be applied to a variety of human figures to avoid discrimination.

4.2 Safety

Based on the OSFA Safety standard, in the section 1910.212,[9] General requirements for all machines, the robotic system should operate under safety regulations. For this project, the biggest issue on that is the pressure of robotic arm as well as the massage head on human's body. This issue would exist among every stage in our project, from design, testing to final demo. To be specific, we have the following safety issues:

1. For the robotic arm and massage head: An inappropriate parameter set for robot motion may hit the user with relatively large force. and the horizontal motion may also hit other people around the workspace of robotic arm. Besides, any wires or sensors may be twisted or broken by the motion of robotic arm. And due to the vibration of massage head, a unreliable mechanical structure may not fix the head stably on the end of robotic arm.

2. For the sensing devices: To sense the force on human's body, we utilize pressure sensor and Ultrasonic distance sensor. The input of those sensors will be used in feedback control of robotic system to adjust the height of robotic massage head to human's body, so that to control the force. The safety issue may come up when one

of those sensor lose connection or don't work properly. A wrong sensor data input may lead to a hazard outcome from robot.

3. For the control unit: An unreliable control program with some inner logical bugs may lead to inappropriate working mode of robotic arm and massage head. Due to the complex factors that may cause the system working in a wrong way, a lack of necessary emergency stop may also lead to serious safety problem when some part of the system raise up problems.

To address those issues, we have the following approaches and testing strategies:

1. Mechanical parts & robotic arms: Some reliability tests should be done on the massage head. We need test the stability of massage head with the motion of robotic arm in different velocity. And an endurance test should also be on the massage head with maximum vibration frequency.
2. Sensing system: All sensors should be fixed tightly on the correct location on the robotic arm or massage head. We will do some tests on those sensors to make sure it could work finely with the motion of the robotic system. Besides, some backup sensor devices could will also setup on the system in case when some devices fail.
3. software & control system: Due to the consideration of accuracy for machine learning models, there should be some accuracy threshold for both both acupoint detection and vital signals processing. We also need to fix all bugs as possible to reduce the program crash risks. And for the processing of data from sensors, we also need to setup a safety threshold to control the motion of robotic arm within a safe range. Moreover, an emergency stop logic should be set when user wants an immediate stop.

References

[1] T. W. Huanbin Gao Shouyin Lu, "Research and development of chinese medical massage robot," *Robot*, vol. 05, pp. 43–52, 2011.

[2] OpenManipulator. "OpenManipulator Pro." (2023), [Online]. Available: https://emanual.robotis.com/docs/en/platform/openmanipulator_p/overview/ (visited on 03/08/2023).

[3] C. Z. Yu Wang, "Novel acupuncture robot," *Robot Technique and Application*, vol. 04, pp. 37–39, 2007.

[4] X. C. Yaonan Dai, "Design and implementation of moxibustion robot with constant temperature on human back spine," *Computer Engineering and Applications*, vol. 09, pp. 222–228, 2019.

[5] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

[6] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[7] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/08/2023).

[8] ACM. "ACM Code of Ethics." (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/08/2023).

[9] OSHA. "OSHA standards for the robotics industry." (2018), [Online]. Available: https://www.osha.gov/robotics/standards.