Báo cáo

**7.11. Thiết kế giao diện RF cho điều khiển từ xa (android/ máy tính bảng)**

Để dễ dàng tương tác với humanoid robot khi robot đang hoạt động qua các thành phần như điều khiển đầu, điều khiển giọng nói, điều khiển tay, điều khiển dáng đi,... Mà không cần phải làm việc trực tiếp với bộ điều khiển trung tâm hay giao diện trên robot, một giao diện điều khiển tổng hợp các tác động là cần thiết và thuận tiện cho người dùng.

Để đồng bộ với hệ thống và không làm phức tạp thêm phần cứng, bộ điều khiển từ xa sử dụng chuẩn truyền không dây Wifi, khi một chức năng được chọn, một node tương ứng sẽ publish topic yêu cầu lên ROS Server, từ ROS Server này, ROS trung tâm sẽ subscribe topic và dọc được message, từ đó, thực hiện yêu cầu như người dùng mong muốn.



**8.8. Thiết kế phần mềm điều khiển robot từ xa qua máy tính bảng.**

Để thiết kế bộ điều khiển robot từ xa, ta cần đặt ra các yêu cầu mà người dùng mong muốn robot có thể đáp ứng, các tác vụ cần điều khiển:

* Đầu:

+ Truyền ảnh trực tiếp từ mắt humanoid đang di chuyển về cho người dùng trên máy tính bảng.

+ Chức năng chụp ảnh.

+ Chức năng nhận dạng khuôn mặt.

+ Chức năng nhận dạng vật thể, màu sắc,...

Các chức năng trên được thực hiện trong ROS trung tâm, tuy nhiên, để robot thực hiện, cần phải kích hoạt bằng tay từ màn hình điều khiển.

+ Chức năng nhận dạng đường đi (đi thẳng, rẽ trái, rẽ phải, dừng lại,…)

* Âm thanh:

+ Nhận dạng giọng nói bằng tiếng anh, thực hiện lệnh.

+ Phát âm từ các audio có sẵn

* Tay:

+ Tác vụ tay như chào, tạm biệt (vẫy tay),..

+ Tác vụ bàn tay như HI, bắt tay, cầm, nắm,...

* Chuyển động:

+ Tác vụ chuyển động cơ bản như đi lên phía trước, rẽ trái, rẽ phải,

+ Mức độ nâng cao như đi nhanh, đi lùi,..

**9.11. Chế tạo giao diện RF cho điều khiển từ xa (android/ máy tính bảng)**

Từ các yêu cầu thiết kế giao diện, để tạo giao diện điều khiển, một trang web html được phát triển, trang web có thể truy cập ở bất kì thiết bị nào (máy tính, máy tính bảng, điện thoại thông minh) hay bất cứ hệ điều hành nào (Windows, Linux, MacOS, Android) miễn là có kết nối Wifi cùng với robot đều có thể điều khiển được robot. Sau khi phát triển, trang web được nhúng vào một ứng dụng android lập trình bằng Android Studio để dễ dàng mở và chạy ứng dụng cũng như điều khiển robot.

Để truy cập camera gắn trên đầu robot, node usb\_cam cần được chạy trong chương trình ROS.

Để ứng dụng có thể mở được hình ảnh mà humanoid đang nhìn, trước tiên, hình ảnh được đẩy lên ROS Webserver bằng node mjpeg\_server.

Để trang web có thể publish một message từ một topic cần có sự hỗ trợ từ node ROS rosbrigde\_websocket (tham khảo ROS)

Giao diện được thiết kế như sau:

Phần trên bao gồm các phần điều khiển Robot như đã nêu, để người sử dụng chọn điều khiển:

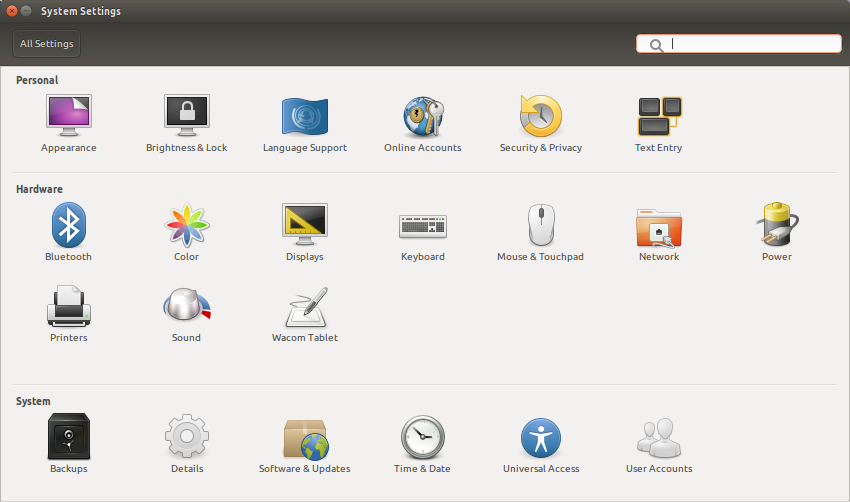
- Điều khiển đầu UXA90.

- Điều khiển UXA90 bằng giọng nói và thu âm cho UXA.

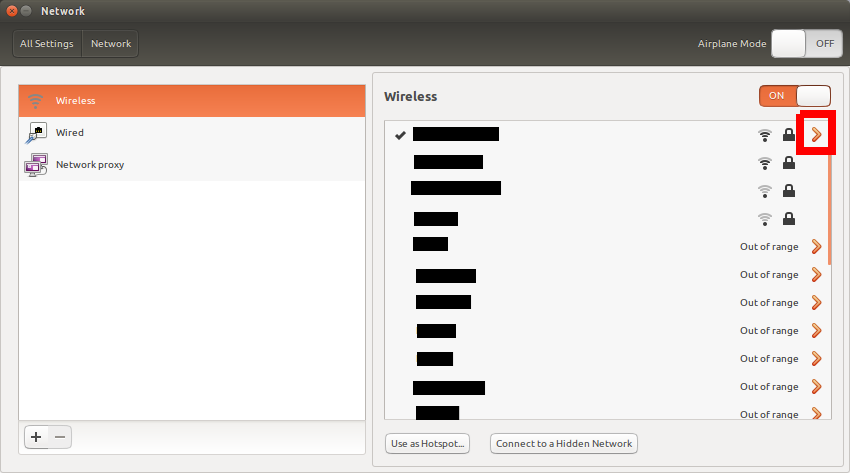
- Điều khiển cánh tay cho UXA90.

- Điều khiển chuyển động cho UXA90.

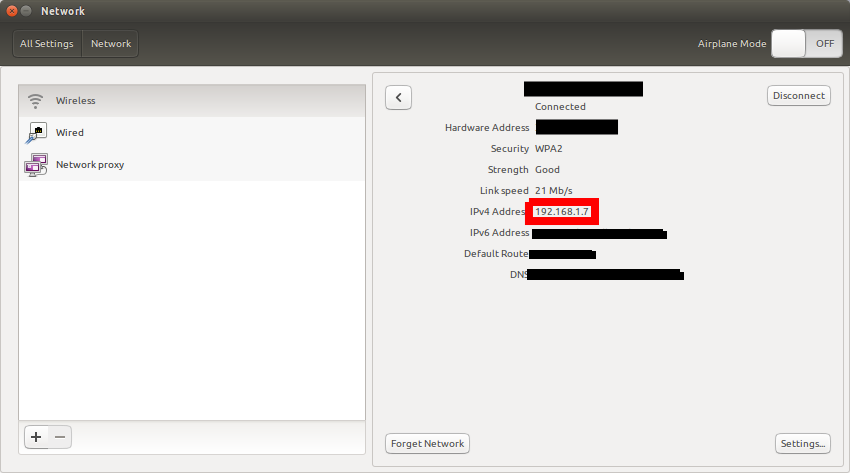
Để kết nối với Robot, đầu tiên ta phải kết nối ứng dụng và robot trong cùng một mạng wifi. Để lấy địa chị IP của Robot, trong hệ điều hành Linux của robot, vào System Settings



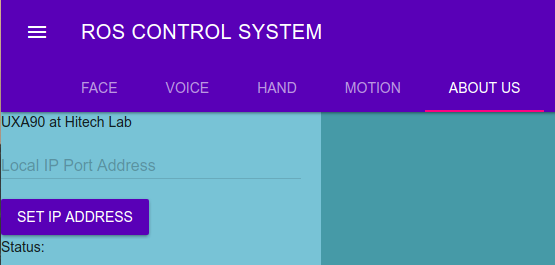
Chọn Networks, vào mạng wifi đang kết nối:



Xem địa chỉ trên IPv4 Address, ví dụ như trong hình là 192.168.1.7. Tên miền 192.168.1 không thay đổi theo mạng wifi, chỉ địa chỉ cuối cùng mới thay đổi theo mỗi lần kết nối.

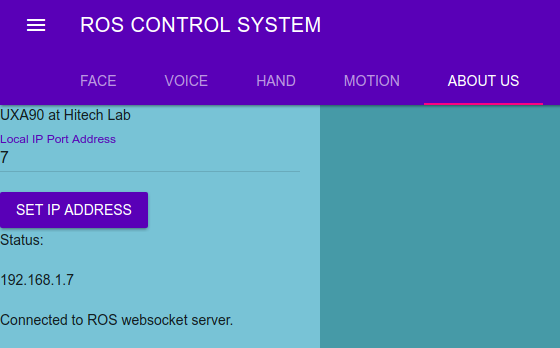


Sau khi biết được địa chỉ IP của robot, vào tab About Us trên giao diện ứng dụng để nhập địa chỉ và kết nối với Robot:



Nhập địa chỉ cuối cùng của IPv4 Address vào textbox, (vd: IP: 192.168.1.7 thì nhập là 7), chọn nút nhấn SET IP ADDRESS để liên kết với Robot.

Khi thành công, giao diện sẽ thể hiện:



Như vậy, ứng dụng đã kết nối thành công với ROS UXA90 Light. Sau đây là các tab điều khiển từng chức năng Robot dùng ứng dụng:

- Điều khiển đầu:

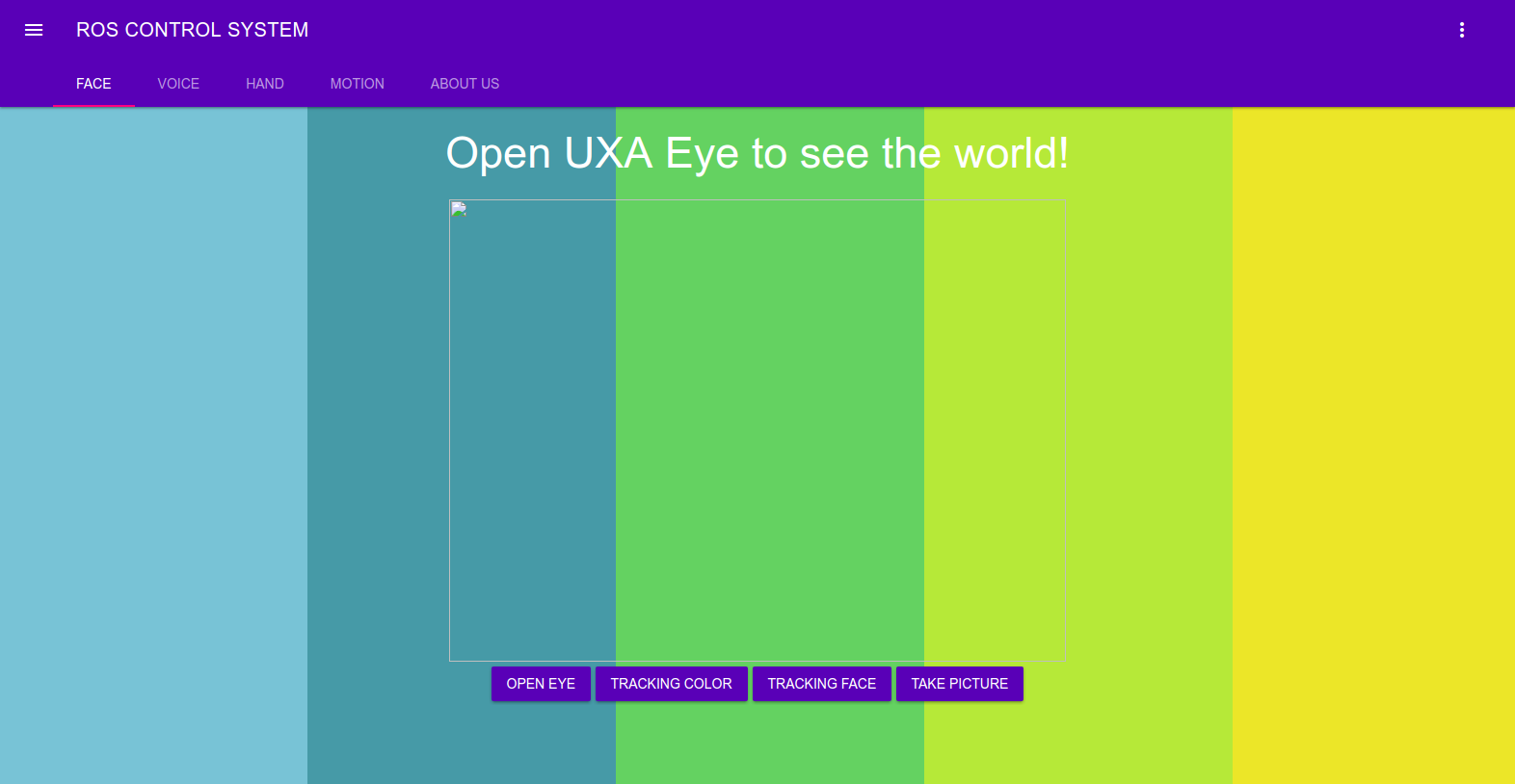
Các phím điều khiển:

+ Open Eye: kết nối ứng dụng với camera trên mắt Robot.

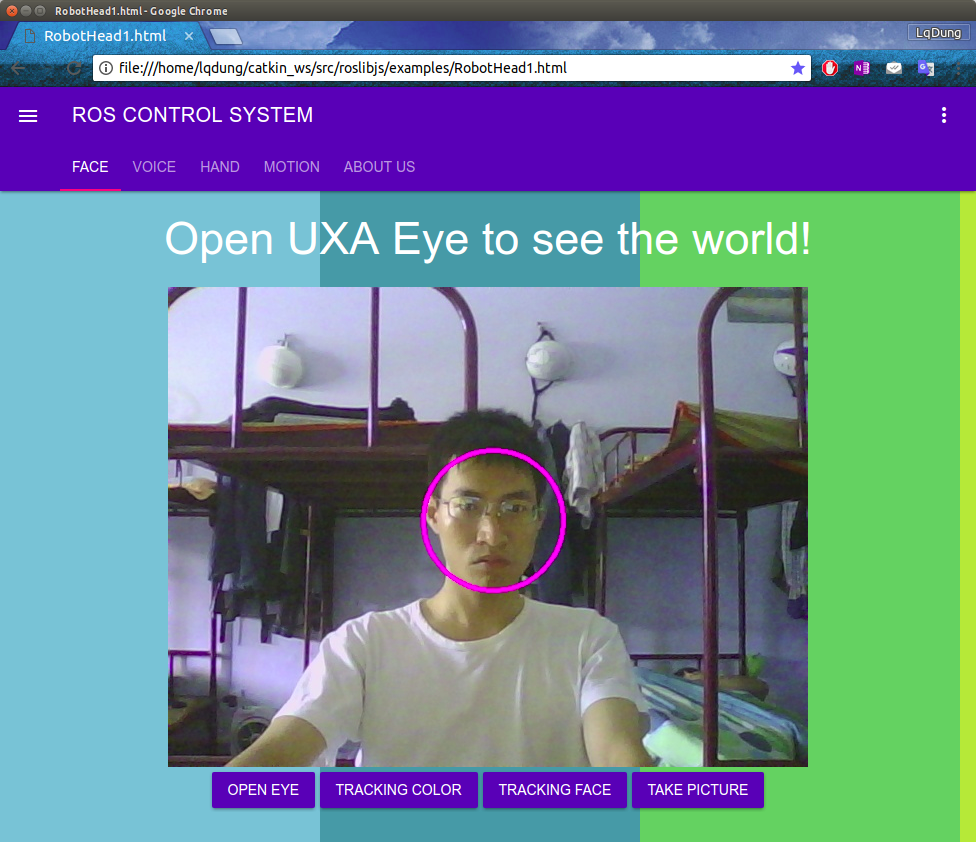
+ Tracking Color: cho mắt Robot bắt theo màu được lập trình trước.

+ Tracking Face: cho mắt Robot bắt theo mặt người gần nhất mà nó nhìn thấy.

+ Take Picture: chụp ảnh.



Demo giao diện ứng dụng khi sử dụng tính năng Tracking Face:



- Điều khiển giọng nói: Robot sử dụng ứng dụng Node PocketSphinx, là node ứng dụng nhận dạng dọng nói được public trên github: <https://github.com/cmusphinx/pocketsphinx,> cho phép lập trình các ứng dụng nhận dạng được câu từ lệnh bằng tiếng Anh được khai báo trước như:

+ Introduce Yourself

+ Move Forward

+ Turn Left

+ Turn Right...

Từ đó, robot có thể biết câu lệnh và thực hiện tương ứng.

Để phát audio từ dữ liệu thu âm có sẵn, Robot sử dụng Node SoundPlay, là node được phát triển và được public trên github: <https://github.com/ros-drivers/audio_common/tree/master/sound_play>, cho phép Robot phát audio các tệp file có đuôi .wav, .ogg,...



- Điều khiển cánh tay:

Các nút chức năng trên ứng dụng:

+ Start: chọn để bắt đầu điều khiển cánh tay

+ Hi: cho cánh tay thực hiện cử chỉ chào người đối diện

+ Shake hand: cho robot thực hiện bắt tay với người đối diện

+ Hold: cho robot thực hiện giữ một vật nhẹ, dễ nắm.

Để thực hiện việc điều khiển, ứng dụng gán mỗi nút chức năng với một ID tương ứng, khi nút được chọn, ID sẽ được phát message đi trong topic “/Hand\_Control”, ROS Webserver cho phép topic truyền đi được gởi đến robot, từ đó robot biến được topic và message cần thực hiện và thực hiện lệnh tương ứng.



- Điều khiển chuyển động:

Tương tự với điều khiển cánh tay, ứng dụng sử dụng topic “\Motion\_Control” để truyền message điều khiển chuyển động robot, các nút chức năng:

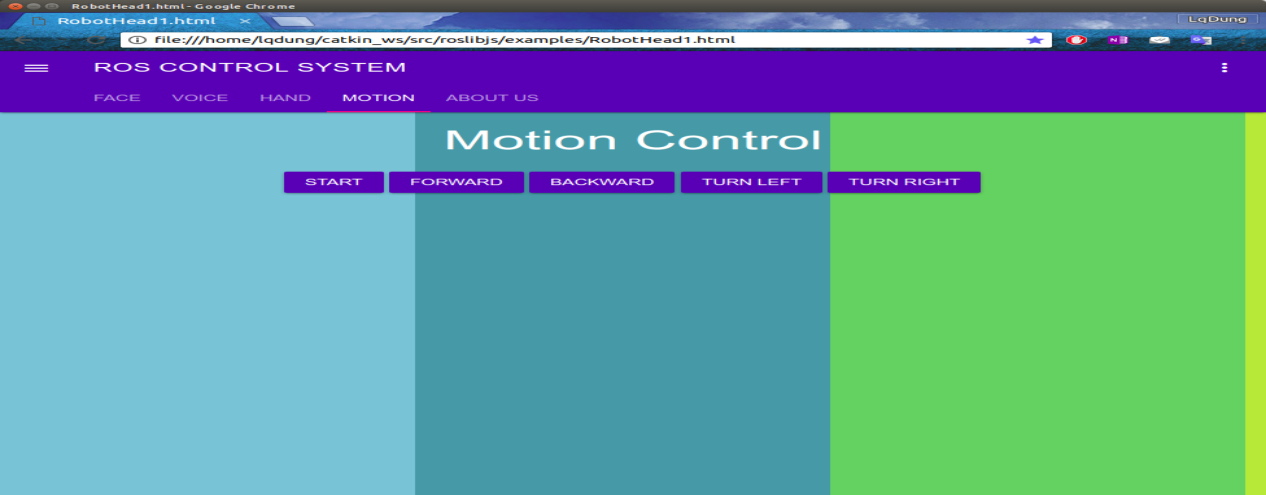
+ Start: bắt đầu điều khiển chuyển động của robot

+ Forward: ra lệnh cho robot di chuyển tới một bước

+ Backward: ra lệnh cho robot đi lùi một một bước

+ Turnleft: ra lệnh cho robot quay sang trái

+ Turn Right: ra lệnh cho robot quay sang phải



Giao diện chương trình sau khi nhúng web vào Android Studio:

