Thuyết trình

- Luồng truyền thông tin

=== Các ứng dụng IVI sẽ gửi thông tin tới home , Home update real time tương ứng lên widget

- Lý do chạy Climate app trc: Vì phải đăng kí Connect(Object-Service) trc

- Đóng mở ap: cơ chế pop/push của Stack

- Cơ chế signal slots(3 dạng chính) C++ to C++, C++ to Qml, Qml to C++

#Signal & Slots:

- Signal tiếng Việt có nghĩa là tín hiệu.Signal không có kiểu trả về, kiểu trả về của signal luôn luôn là void.

- Slot chẳng qua cũng là một phương thức bình thường của một lớp, các phương thức này sẽ được gọi khi có một signal nào đó được phát đi.

- Signal và slot được kết nối qua từng đối tượng (chứ không phải qua từng lớp như nhiều bạn vẫn nghĩ).

Tức là chúng ta chỉ có kết nối đối tượng này với đối tượng kia chứ không kết nối lớp này với lớp kia

- Một signal có thể kết nối đến nhiều slot và một slot có thể kết nối đến nhiều signal.

- Slot sẽ được gọi khi có signal tương ứng được phát ra

- Để phát các signal thì bạn dùng từ khóa emit.

- Dùng QObject::connect để kết nối sign với slot

==> thuận tiện cho việc gọi hàm

#QObject:

- QObject là trung tâm của Mô hình Đối tượng Qt.

Tính năng trung tâm của mô hình này là một cơ chế rất mạnh mẽ để giao tiếp đối tượng liền mạch được gọi là signal và slot.

- QObject là lớp cơ sở của tất cả các đối tượng Qt

- Model của QObject lưu trữ theo dạng cây

#QAtractListModel:

- QAbstractListModel cung cấp một mô hình trừu tượng có thể được phân lớp để tạo mô hình danh sách một chiều

- QAbstractListModel cung cấp giao diện tiêu chuẩn cho các model thể hiện dữ liệu của chúng dưới dạng một chuỗi các mục không phân cấp đơn giản.

Nó không được sử dụng trực tiếp mà phải được phân lớp.

#QProperty:

- Thực hiện ràng buộc giữa các thuộc tính trong chương trình

- Có các chức năng READ, WRITE, RESET, NOTIFY, MEMBER...

#Q\_invoke:

- Đánh dấu ở trong 1 phương thức sẽ có 1 phương thức chạy trong nó

#Pointẻ:

- Một con trỏ (a pointer) là một biến được dùng để lưu trữ địa chỉ của biến khác

- Khác với tham chiếu, con trỏ là một biến có địa chỉ độc lập so với vùng nhớ mà nó trỏ đến

#D-bus:

Là 1 IPC (inter-process communication) kết nối app với nhau

- Dùng connect home app với climate app (tương tự như hệ thông client-sever) để chia sẻ thông tin

- CUng cấp cho hệ thông linux

- Tạo ra các adaptor và các bus

- Để tạo connect: phải có Service name(người vận chuyển), object path(địa chỉ nhận), interface(gửi theo đường nào)

Cách làm với dbus:

- Tạo file xml: dbuscpp2xml <filename> copy vào 2 app

- Config += c++11 dbus để sử dụng dbus

- khai báo Adaptor ở .pro của climate app gửi

- khai báo interface ở .pro của home app nhận

- Sau đó run qmake để tạo ra adaptor .cpp và .h

- include file adaptor để sử dụng

- Chạy climate app để đăng kí interface và service

- Run qmake homeapp để tạo interface .h .cpp

- Sang home app tạo climatemodel include interface.h

- Nhận connect từ climate app ở climatemodel.cpp

- Nhận signal datachanged add vào đưa vào đối tượng m\_climate

- Khi thay đổi đối tượng ở climate app sẽ lưa lại và setData và signal datachanged sẽ truyền dữ liệu cho nhau

- Update lên widget từ đối tượng m\_climate