4차 산업 주축 분야 중 하나인 사물인터넷을 이용하는 스마트홈의 유일한 물리적인 출입통로인 문을 관리하는 사물의 필요성이 대두되어, 스마트 도어록을 주제로 선정하고 개발하게 되었다. 아두이노의 NFC, 웹캠 모듈을 HTTP5 웹소켓 프로토콜을 이용하여 원격에서 조작할 수 있게 하였다. NFC 카드만을 이용하여 문을 개폐하는 기능을 제공하며, 동시에 UID를 기본 키로 하는 개폐시간 로그 저장, 유저 테이블의 저장된 UID와의 대조가 이루어진다. 도어록에서 물리적으로 카드 등록을 할 수 있으며, 스마트폰 어플에서 원격 문 개폐, 선택한 기간의 로그 조회, 유저 테이블의 이름, 권한, 아이디, 패스워드를 조회/수정, 집 앞 웹캠 스트리밍 서비스를 이용할 수 있다. 스마트폰 어플 로그인은 아이디, 비밀번호만으로 가능하게 로그인 테이블을 따로 가지고 있다. 서버를 거치는 모든 클라이언트들은 웹소켓 프로토콜을 이용하게 하여, 서버에서 하나의 프로토콜에 대한 응답만 관리하는 이점이 있고, 스트리밍은 독자적인 웹서버를 이용하여 서버의 부하를 줄였다. 기존 도어록보다 저렴한 모듈을 부착만 시킴으로서 위의 모든 기능을 이용 가능하게 하였다.

The need for objects managing doors, the only physical access passage of smart homes using the Internet of Things, one of the main areas of the fourth industry, emerged, so the smart door locks were selected and developed as the theme. NFC and webcam modules of Aduino can be operated remotely using HTTP5 WebSockets protocol. It provides a function to open and close doors using NFC cards only, and at the same time, UID is the primary key, and user table is compared to stored UIDs. You can physically register your card from a door lock, open and close remote doors on a smartphone app, view logs for the selected period, view/modify the user table's name, authority, ID and password, and use the webcam streaming service in front of your home. Smartphone application logins have separate login tables that can be done only with ID and password. All clients going through the server have the advantage of using the WebSocket Protocol, managing only one protocol response from the server, and streaming has reduced the server's load using its own web server. All of the above functions are made available by simply attaching modules that are cheaper than conventional door locks.

키워드 – 스마트 도어락, 홈 네트워크, IoT, NFC, 웹 서버, 웹캠 스트리밍, 스마트폰 App

서버쪽 기술들

Sqlite3 모듈 - SQLite는 별도의 서버 프로세스가 필요 없고 SQL 질의 언어의 비표준 변형을 사용하여 데이터베이스에 액세스할 수 있는 경량 디스크 기반 데이터베이스를 제공하는 C 라이브러리입니다. 일부 응용 프로그램은 내부 데이터 저장을 위해 SQLite를 사용할 수 있습니다. 서버 프로그램은 유저 테이블, 로그인 테이블, 로그 테이블을 저장하고 관리하기 위해 이 모듈을 사용하였습니다. 데이터 베이스에 연결 후 Cursor 객체의 execute() 함수를 이용하여 SQL 명령을 수행할 수 있습니다. 저장된 데이터는 영구적이며 이후 세션에서 사용할 수 있습니다. 일반적으로 SQL 연산에 파이썬 변수의 값을 사용할 수 있습니다.

Asyncio 모듈 - asyncio는 async/await 구문을 사용하여 동시성 코드를 작성하는 라이브러리입니다. 고성능 네트워크 및 웹 서버, 데이터베이스 연결 라이브러리, 분산 작업 큐 등을 제공하는 여러 파이썬 비동기 프레임워크의 기반으로 사용됩니다. 파이썬 코루틴들을 동시에 실행하고 실행을 완전히 제어할 수 있습니다. 이벤트 루프는 스레드(일반적으로 주 스레드)에서 실행되며 그 스레드에서 모든 콜백과 태스크를 실행합니다. 태스크가 이벤트 루프에서 실행되는 동안, 다른 태스크는 같은 스레드에서 실행될 수 없습니다. 태스크가 await 표현식을 실행하면, 실행 중인 태스크가 일시 중지되고 이벤트 루프는 다음 태스크를 실행합니다. 서버 프로그램의 클라이언트에 대응하는 모든 함수는 async 구문을 사용하여 동시성 코드로 작성되었습니다. 한 클라이언트와의 연결이 스레드를 너무 오래 점유하지 않도록 함수의 통신 부분에 await 표현식을 사용하여 태스크들의 스레드 점유를 골고루 하게 하였습니다.

Websockets 모듈 - Websockets는 파이썬에서 웹소켓 연결을 관리하는 가장 좋은 코루틴 기반의 API를 제공하기 위해 만들어졌습니다. 정확성과 단순성에 중점을 둔 Python으로 WebSocket 서버 와 클라이언트를 구축하기 위한 라이브러리입니다. Python의 표준 비동기 I / O 프레임워크인 asyncio 위에 구축된 프로그램은 우아한 코루틴 기반 API를 제공합니다. 서버의 모든 프로토콜은 웹소켓을 통해 이루어집니다. 웹소켓의 send(), recv() 함수는 완벽한 await 표현식을 제공합니다.