

마이크로 프로세서 및 실험 Report

2018-1학기 최종 보고서



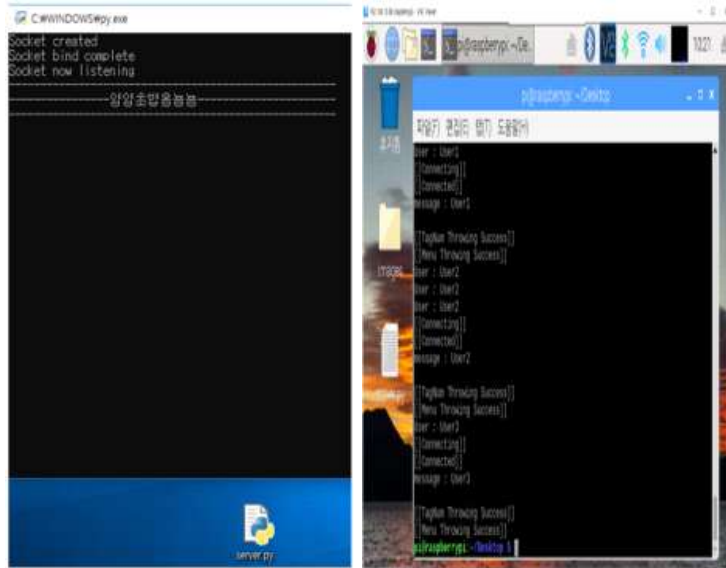
제출일	2019.06.14
담당교수	김영천
전공	IT정보공학과
분반/조	4분반 / 2조
팀장	201511033 최세진
구성원1	201646201 공지나
구성원2	201511019 박경현
구성원3	201716919 서은빈

0. 목차	<p>1. 프로젝트 소개 - - - - - 2p</p> <p>2. 기대 효과 - - - - - 2p</p> <p>3. 구현 계획 - - - - - 3p</p> <p>4. 역할 분담 - - - - - 3p</p> <p>5. 제작 일정 - - - - - 3p</p> <p>6. 제작 과정 - - - - - 3p</p> <p>7. 최종 구현 기능 - - - - - 5p</p> <p>8. 최종 구동 화면 - - - - - 6p</p> <p>9. 문제 해결 - - - - - 6p</p> <p>10. 고찰 - - - - - 7p</p>
1. 프로젝트 소개	<p>▶ KIOSK란</p> <p>- 정보서비스와 업무의 무인자동화를 위하여 대중들이 쉽게 이용할 수 있도록 공공장소에 설치한 무인단말기.</p> <p>▶ 주요 기능</p> <p>- 음식점 태블릿 키오스크 구현 예정</p> <ul style="list-style-type: none"> - 메뉴 안내 - 주문 기능 - 결제 기능 - 부가 기능(챗봇 등) <div data-bbox="1066 790 1254 1193" data-label="Image"> </div> <p style="text-align: center;">키오스크 예시</p>
2. 기대 효과	<p>▶ 편의성 / 활용성</p> <div style="display: flex; justify-content: space-around;"> <div> <p>“60%”</p> <p>패스트푸드 기업 M사의 키오스크 설치 비율</p> </div> <div> <p>“100%”</p> <p>패스트푸드 기업 K사의 키오스크 설치 비율</p> </div> </div> <ul style="list-style-type: none"> - 인건비 절약 - 고객들의 부담 없는 정보 탐색 - 약간의 수정으로 음식점 뿐만 아니라 다양한 분야에서 사용 가능 <p>▶ 비용 절감</p> <div style="display: flex; justify-content: space-around; align-items: flex-end;"> <div style="text-align: center;"> <p>100%</p> <p>한국 A社 Kiosk K-170D 1,228,500W</p> </div> <div style="text-align: center;"> <p>51%</p> <p>미국 A社 Tablet PC-Air 3rd gen 629,000W</p> </div> <div style="text-align: center;"> <p>6%</p> <p>Our Device Raspberry Pi 3B + 7" Touch screen 76,000W</p> </div> </div>

3. 최초 구현 계획	<ul style="list-style-type: none"> - UI 설계 - 서버와 데이터 송·수신 - 개인 ID카드 인식 - 그 외 부가기능 구현
4. 역할 분담	<div> <div> <div>팀장</div> <div>01</div> <div>최세진</div> </div> <div> <div>02</div> <div>공지나</div> </div> </div> <div>:UI 구현</div> <div> <div>03</div> <div>박경현</div> </div> <div> <div>04</div> <div>서은비</div> </div> <div>:서버 & ID카드 구현</div>
5. 제작 일정	<p>프로젝트 계획 및 준비 4/16 내 및 서버 구현 시작 6/2 프로젝트 마무리 6/11</p> <p>4/4 중간 발표 4/29 기본 기능 구현 완료 & 최적화 시작 6/7 최종 발표</p>
6. 제작 과정	<p>▶ 서버 & 클라이언트 부분</p> <p>I. Python을 이용하여 서버 및 클라이언트 기초 구현</p> <div> <pre> 1 #!/usr/bin/env python 2 # coding: utf-8 3 4 import socket 5 6 HOST = '192.168.10.89' 7 PORT = 8888 8 9 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) 10 s.connect((HOST, PORT)) 11 while 1: 12 msg=input() 13 s.sendall(msg.encode(encoding='utf-8', errors='strict')) 14 data = s.recv(1024) 15 print('result: ' + data.decode()) 16 s.close() </pre> </div> <div> <pre> 1 #!/usr/bin/env python 2 # coding: utf-8 3 4 import socket 5 6 HOST, PORT = '192.168.10.89', 8888 7 8 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) 9 s.bind((HOST, PORT)) 10 s.listen(10) 11 print('Socket bind complete') 12 13 #3. Listen for incoming connections 14 s.listen(10) 15 print('Socket now listening') 16 while 1: 17 #4. Accept connection 18 conn, addr = s.accept() 19 print('###Connected with: ' + addr[0]+' '+' '+str(addr[1])) 20 21 #5. Read/Send 22 data = conn.recv(1024) 23 print(data) 24 if conn: 25 conn.sendall(data) 26 print('### 전송: ' + data.decode()) 27 28 data = conn.recv(1024) 29 if conn: 30 conn.sendall(data) 31 print('### 전송: ' + data.decode()) 32 result = data.decode() 33 smp = result.split() 34 smp[0] = smp[0].upper() 35 for i in range(0, len(smp)-1): 36 print(smp[i]+' ') 37 print('### 전송') 38 conn.close() 39 s.close() </pre> </div>

7. 최종 구현 기능

▶ 서버 & 클라이언트 부분



서버(Win10)에서 구동 클라이언트(Rasberry Pi) 구
화면 동 화면

1. Server

- 작동 상태로 대기
- 클라이언트가 메시지를 보내면 화면에 띄우기
- User를 구분하여 정상적으로 받았음을 클라이언트에게 전송

2. Client

- 사용자가 주문한 내역을 GUI Program으로부터 수신
- 주문 내역 항목을 하나의 문자열로 합친 뒤 서버로 전송
- 서버에서 정상적으로 수신했다는 메시지를 받으면
주문완료 메시지를 화면에 띄우도록 GUI Program에 요청

▶ 터치스크린 GUI부분



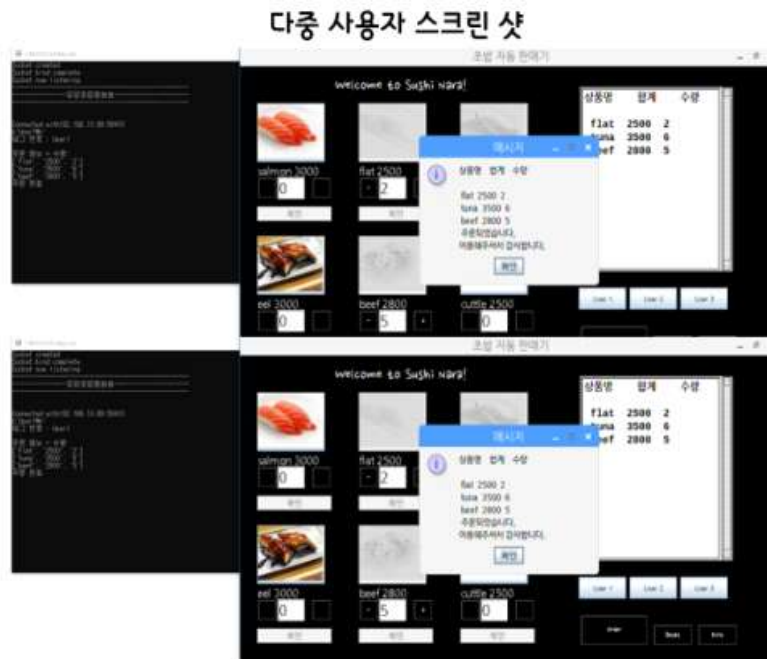
1. Interface

- 7" 터치스크린에 맞는 아이콘 크기 조정
- 음식을 고르고, 수량을 정하고, 확인을 누르는 직관적 UI
- Java Swing을 사용 -> 다른 기기에서도 JVM만 있으면 동일한 UI제공

2. Source Code

- 여러 Open Source 연구 후 라즈베리파이에 어울리도록 직접 설계
- WindowBuilder 사용으로 통일된 코드 디자인
- Clident 내장으로 프로그램 내에서 서버와 통신할 수 있음

8. 최종 구동 화면



9. 문제 해결

1. Server-Client는 Python으로, GUI는 JAVA로 구현

- Client를 JAVA로 변환시켜 GUI Program에 내장

2. 클라이언트 -> 서버 전송 데이터 값에서 태그번호와 주문내역이 섞여버림

- 클라이언트 전송 시 구분자 사용 등의 방법으로 조치

3. Raspberry Pi와 JVM과 Windows 개발 환경의 JVM버전이 달라 GUI 프로그램이 Pi에서 실행되지 않음.

- Windows의 JDK를 구버전으로 다운그레이드 함

4. RFID 사용자 인식을 목표로 삼았으나, 현실적 제약으로 인한 구현 불가

- 대안으로 스마트폰을 NFC 리더기로 활용하려했으나, 수시적으로 종료되어 S/W적으로 User구분 구현

10. 고찰

▶ 서은빈

파이썬과 자바를 기반으로 한 서버와 클라이언트 사이의 소켓 통신을 바탕으로 라즈베리파이를 활용한 태블릿 키오스크로 제작함으로써 실생활에서 접해보았던 기기를 실제로 구현해 볼 계기가 되었다. 프로젝트를 통해 구현해보았던 소켓 통신과 UI등을 변형하면 여러 분야에 걸쳐 응용이 가능하기 때문에 도움이 되었던 경험이었다.

▶ 박경현

다른 장치들끼리 소통하는 방식 중 가장 유용하게 쓰이는 기법중 하나가 소켓 프로그래밍을 이용하여 정보를 전달하는 방식입니다. 이 기능을 가지고 키오스크 제작을 함으로서 다른 장치간 정보전달 하는 방식에 대해 공부하게 되었고, 다음에 프로젝트 할 때도 지금 배운 내용을 사용할 수 있을 것 같습니다.

▶ 공지나

프로젝트를 하면서 원활하게 잘될 때도 있었지만 문제가 생길 때마다 팀원들과 함께 그 문제점을 해결하려고 노력했습니다. 또한, 팀원들과 의견 조율하는 데 있어 서로의 의견을 존중해주며 어려움 없이 진행할 수 있어 좋았습니다.

▶ 최세진

팀 프로젝트를 하게 되면 개인 프로젝트와는 많은 차이가 있고, 그 차이점에서 나오는 단점을 극복하며 장점을 극대화하는 것이 팀장의 일이다 생각하며 프로젝트에 임했습니다. 터치 스크린에 맞는 GUI 구현은 처음이었지만, 팀원과의 소통과 협업을 통해 잘 해결하였고, 성공적인 프로젝트 결과물을 만들어 냈다고 생각합니다. 더욱 발전하고 완성도 높은 프로젝트 기획 및 진행을 하는 데 한 걸음 더 다가가는 계기가 되었습니다.