

A Development of Shoes Cleaner Control System using Raspberry Pi

Deukchang Hyun*

*Professor, Dept. of AI Computer Engineering, Far East University, Eumsung, Korea

[Abstract]

Since leather shoes cannot be washed with water, there is a need for a cleaning method that can remove extraneous substance from the inside and outside of shoes and sanitize the inside of shoes without using water. For this purpose, this paper develops a shoes cleaning machine control system that automatically controls the entire process of shoes cleaning in a shoes cleaning machine that quickly cleans the inside and outside of shoes using compressed air, sterilization solution. The developed system uses Raspberry Pi, a general purpose single board computer(SBC), to control various actuators of the shoes cleaning machine. The shoes cleaning machine operated by the developed system shows a sterilization efficiency of more than 99% and an odor removal efficiency of more than 86% in a cleaning time of less than 1 minute.

▶ **Key words:** IoT, embedded computing, Raspberry Pi, shoes cleaner, single board computer

[요 약]

가죽제품 신발은 물 세척을 할 수 없기 때문에 물을 이용하지 않으면서 신발 내·외부의 이물질을 제거하고, 신발 내부를 살균할 수 있는 세척 방법이 필요하다. 이를 위해 이 논문에서는 물을 사용하지 않고 고압의 압축공기와 살균 약액 및 자외선 램프를 이용하여 신발 내·외부를 신속하게 세척하는 신발세척기에서 신발 세척의 전 과정을 자동으로 제어하는 신발세척기 제어 시스템을 개발한다. 개발 시스템은 범용 단일보드 컴퓨터(SBC:single board computer)인 라즈베리파이(RaspberryPi)를 이용하여 신발 세척기의 각종 구동 장치들(actuators)을 제어한다. 개발 시스템에 의해 운영되는 신발 세척기는 1분 이내의 세척 시간에 99% 이상의 살균력과 86%이상의 악취제거 효율을 보여준다.

▶ **주제어:** 사물인터넷, 임베디드 컴퓨팅, 라즈베리파이, 신발 세척기, 단일보드 컴퓨터

-
- First Author: Deukchang Hyun, Corresponding Author: Deukchang Hyun
 - *Deukchang Hyun (hyundc@hanmail.net), Dept. of AI Computer Engineering, Far East University
 - Received: 2024. 01. 29, Revised: 2024. 07. 08, Accepted: 2024. 07. 09.

I. Introduction

가죽을 소재로 하는 신발은 물을 사용한 세척을 할 수 없기 때문에 신발의 청결 유지에 어려움이 있다. 이 문제를 해결하기 위해 압축공기와 살균 약액을 이용하여 신발 내부의 깊숙한 부분까지 청결하게 세척할 수 있는 신발 세척기를 개발하였다. 개발된 신발 세척기는 세척 조건에 따라 신발 세척기의 상태를 감지하고, 이를 반영하여 신발 세척 과정에서 작동하는 각종 기계/기구들을 자동으로 제어하는 제어 시스템이 필요하다.

세척 조건은 최적의 세척 결과를 도출하기 위해 압축공기의 공급 시간, 약액의 공급 시간 및 총 세척 시간 등을 규정한 것으로서 여러 번의 실험 결과 얻은 파라미터들로 구성된다. 신발 세척기 제어 시스템은 이 세척 조건을 자동으로 수행할 수 있게 하는 시스템으로서 각종 기계/기구에 대한 제어 수행 기능과 상황 파악을 위한 센싱 정보 입력 및 해석 기능을 갖추고 있어야 한다.

이 논문에서 구현하는 신발 세척기 제어 시스템은 제어 엔진과 제어 구동부로 구성된다. 제어 엔진은 제어 조건에 따라 제어 신호를 생성하여 제어 구동부에 전송하거나 제어 구동부로부터 수신되는 센싱 정보를 해석하고 대응하는 역할을 수행한다. 제어 구동부는 제어 엔진으로부터 전송되는 제어 신호에 대응한 릴레이 구동을 통해 장치의 작동을 제어한다. 이 중에서 제어 엔진은 사물인터넷을 위한 오픈 소스 하드웨어 플랫폼을 이용하여 구현한다.

이를 위한 논문의 구성은 다음과 같다. 먼저 2장에서는 관련연구를 기술한다. 3장에서는 연구의 도메인인 신발 세척기를 고찰하고, 신발세척기를 최적으로 제어하기 위한 제어 시나리오를 도출 한다. 4장에서는 3장에서 도출한 제어 시나리오에 따른 제어 회로의 구성과 제어로직의 구현에 대한 사항을 기술하고, 5장에서는 구현 시스템에 대한 평가 결과를 기술한다. 마지막으로 6장에서 결론 및 향후 연구에 대해 논한다.

II. Related works

오픈 소스 하드웨어 플랫폼은 사물 인터넷(IoT : internet of things) 및 임베디드 컴퓨팅(embedded computing) 분야에서 통신과 제어를 위해 광범위하게 활용되는 장치이다[1]. [2]에서는 7가지 장치 및 파생 장치들을 포함하여 전반적인 특성을 분석하고 있다.

[2]에서 요약하는 장치들 중에 주목할 장치로 라즈베리 파이(RaspberryPi), 아두이노(Arduino), ESP8266을 들 수 있다. 아두이노와 ESP8266은 CPU와 통신 프로토콜을 내장하고, 다수의 범용 입출력(GPIO: general purpose I/O)핀을 갖추고 있어서 IoT 또는 임베디드 컴퓨팅 응용 개발을 위한 강력한 도구로 활용된다[3, 4]. 특히 ESP8266은 ESP32로 업그레이드 되면서 WiFi, BLE(blueetooth low energy), Zigbee가 통합 제공되는 버전도 있기 때문에 IoT 응용에 더 적합한 플랫폼으로 성장하고 있다[4]

ESP32가 아두이노 보다 훨씬 우수한 기능과 성능을 갖추고 있지만, 광범위한 레퍼런스를 갖추었을 뿐만 아니라 더 큰 개발자 커뮤니티를 구축하고 있다는 점에서 아두이노가 더 선호되는 상황이다[3]. 그럼에도 불구하고 아두이노 또는 ESP32는 범용의 OS를 지원하지 않기 때문에 SBC로 분류할 수 없다. 반면에 라즈베리파이의 경우 데비안 계열의 리눅스 OS인 Raspbian OS를 탑재하고 있을 뿐만 아니라 Windows를 구동할 수도 있는 독자적 컴퓨팅 환경을 제공하는 등 대표적인 SBC 기종이라 할 수 있다[1].

이 논문에서 개발하는 신발세척기 제어 시스템은 신발 세척기의 작동 과정을 제어하는 기능뿐만 아니라 사용자 인터페이스의 추가 구현이 필요하기 때문에 SBC인 라즈베리 파이와 제어 엔진의 플랫폼으로 더 적합하다.

이와 같은 관점에서 라즈베리파이를 활용한 사례로 [5]에서는 의수(prosthetic hand)의 정확한 작동을 학습하기 위한 학습모듈 개발에 활용하였고, [6]에서는 저비용으로 대규모 클러스터 구성을 통한 슈퍼컴퓨팅 구현을 소개하고 있다.

라즈베리파이의 가장 대표적인 응용 분야는 임베디드 컴퓨팅 분야로서 [7]에서는 임베디드 컴퓨팅 교육에 관한 내용을 다루고, [8, 9, 10]에서는 이미지 처리(image processing) 또는 컴퓨터 비전(computer vision) 분야를, [11, 12]에서는 머신러닝분야에서 라즈베리파이를 활용하였으며, [13, 14]는 신경망(neural network)응용의 실행환경으로서의 라즈베리파이를 다룬다.

또한 이 논문과 유사한 도메인으로서 라즈베리파이를 제어 시스템으로 활용하는 사례인 [15]에서는 인터넷을 이용하여 모바일 로봇을 감시하고 제어하는 플랫폼으로 활용하고 있으며, [16]은 라즈베리파이를 PLC(programmable logic control)를 대체하는 산업용 컨트롤러로 개발한 내용에 대해 소개 한다.

III. Shoes cleaner analysis and control scenarios deriving

1. shoes cleaner analysis

이 연구의 대상이 되는 신발 세척기는 물을 사용하지 않고 압축공기를 이용하여 신발 내부 이물질들을 제거하고, 신발 내부에 번식하는 세균류 살균을 위해 약액을 이용하여 살균·소독하는 기능을 구현한 장치이다. 다음의 Fig. 3.1은 신발 세척기의 구조를 보여주는 그림이다.

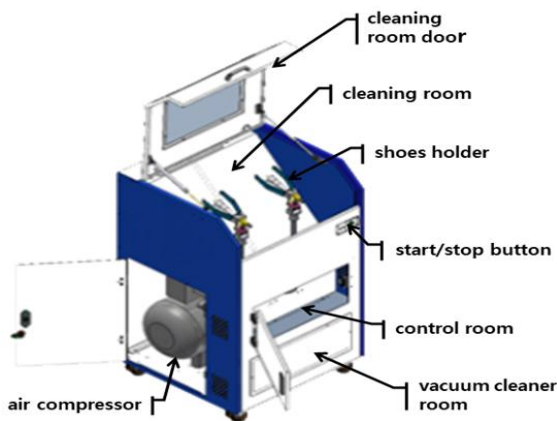


Fig. 3.1. shoes cleaner shape

Fig 3.1에서 세척실 내부의 신발 거치대(shoes holder)에 신발을 거치하고, 도어를 닫은 다음 시작/종료 버튼(start/stop button)을 누르면 세척을 시작한다. 다음의 Fig. 3.2는 신발 세척기에서 세척이 이루어지는 원리를 보여주는 그림이다.

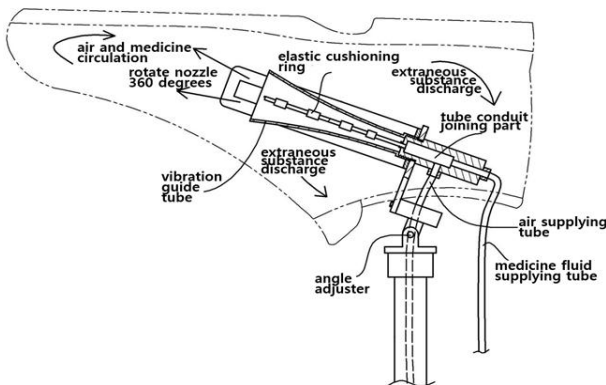


Fig. 3.2. principles of shoes cleaning

Fig. 3.2에서 압축공기 공급관(air supplying tube)을 통해 공급되는 고압의 압축공기 압력과 속도로 인해 노즐 끝에서 강한 회전이 발생하여 회오리 효과를 만들어낸다.

그리고 이 회오리 효과로 인해 신발 내부의 이물질들을 효과적으로 털어낼 수 있다. 한편 약액 공급관(medicine fluid supplying tube)을 통해 공급되는 약액은 합류관로부(tube conduit joining part)에서 고속의 압축공기 흐름에 의해 발생하는 베르누이 원리(Bernoulli theorem)에 의해 신발내부에 살포되도록 하여 신발 내부 세균 살균에 작용하도록 한다.

2. shoes cleaner control scenarios

신발 세척기의 작동은 신발 내부 이물질 제거를 위한 압축공기 토출, 신발 내부 살균을 위한 약액 토출, 신발 외부 살균을 위한 uv 램프 작동, 신발 내부에서 탈거된 이물질 집진용 진공청소기 가동, 신발 세척기가 작동중임을 시각적으로 표시하는 LED 램프 점등, 도어 개폐여부에 따른 작동 상태 제어가 가능해야 한다. 이를 반영한 신발 세척기 작동 조건은 다음의 Table, 3.1과 같다.

Table 3.1. shoes cleaner operation condition

working part	start time	ending time	explanation
air discharge	0	10	high-pressure air blast to remove extraneous substance
sanitizer discharge	0	3	optimal discharging time to maximize sanitizing effectiveness
vacuum cleaner	0	60	operation for the entire shoes cleaning period
uv lamp	0	60	operation for the entire shoes cleaning period
LED light	0	60	operation for the entire shoes cleaning period

Table. 3.1에서 시작시간(start time)과 종료 시간(ending time)의 단위는 초(second) 단위이다. 그리고 Table. 3.1의 신발 세척 조건에서 핵심은 압축공기와 살균 약액의 공급시간이다. 이 두 가지 조건이 신발 내부의 세척에 결정적으로 영향을 미치는 요인이기 때문이다. 최적의 신발 세척을 유도하기 위해 여러 번의 실험 및 평가를 통해 압축공기의 공급 시간을 10초로 설정하고, 약액 공급 시간을 3초로 설정하였다.

Table. 3.1의 작동 조건을 반영한 신발 세척 과정은 다양한 상황에 대응해서 항상 정확하게 작동하기 위한 작동 시나리오가 필요하다. 다음의 절 에서 추정 가능한 모든 상황에 대응하는 시나리오를 제시한다.

2.1 normal state scenario

신발 세척을 위한 첫 번째 시나리오는 예외 상황 없이 신발 세척이 진행되는 시나리오으로써 다음의 Fig. 3.3과 같이 모델링할 수 있다.

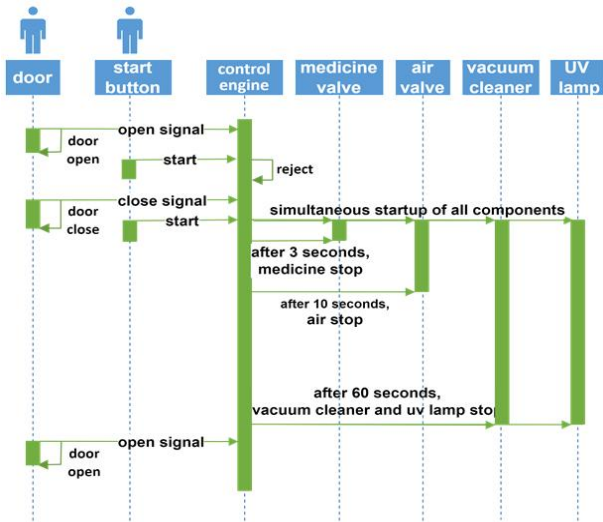


Fig. 3.3. shoes cleaner operation under normal condition

Fig. 3.3에서 신발 세척기의 작동은 도어 개폐여부를 확인하는 절차부터 시작한다. 신발이 신발 거치대에 장착된 상태에서 도어가 닫혀 있음을 확인한 다음 작동을 시작하며, 압축공기와 약액 토출, 진공청소기, 자외선 살균 램프 등이 Table. 3.1의 조건에 따라 동시에 시작하고 순차적으로 종료한다.

2.2 door open during shoes cleaning

Fig. 3.3의 시나리오에서, 신발 세척이 완료되지 않은 상태에서 도어가 열리는 예외 상황이 발생할 수 있으며, 이 경우 모든 구동 장치들의 작동을 중단해야 한다. 다음 Fig. 3.4는 이 상황을 반영한 시나리오이다.

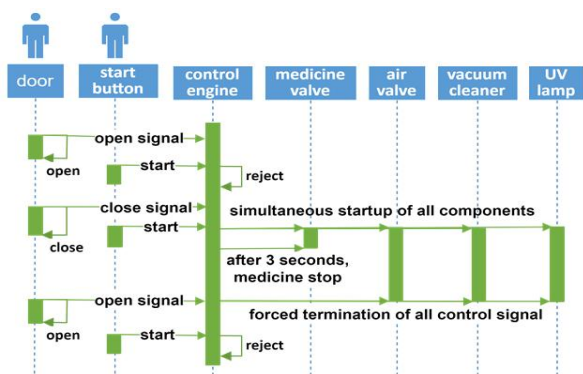


Fig. 3.4. door open during shoes cleaning

Fig. 3.4에서 확인할 수 있듯이 신발 세척 진행 중 도어가 열리면 모든 구동장치들이 작동을 멈춤으로써 신발 세척을 강제적으로 종료한다.

2.3 shoes cleaning start while the compressor running

압축공기 생산을 위한 컴프레서는 압축 탱크의 압력 게이지에 따라 자동으로 가동된다. 따라서 컴프레서는 신발 세척 절차와는 무관하게 자율적으로 작동하며, 신발 세척기는 이를 고려하여 세척 절차를 수행할 수 있도록 해야 한다. 다음의 Fig. 3.5는 컴프레서의 작동 중에 신발세척을 시작하는 시나리오를 보여준다.

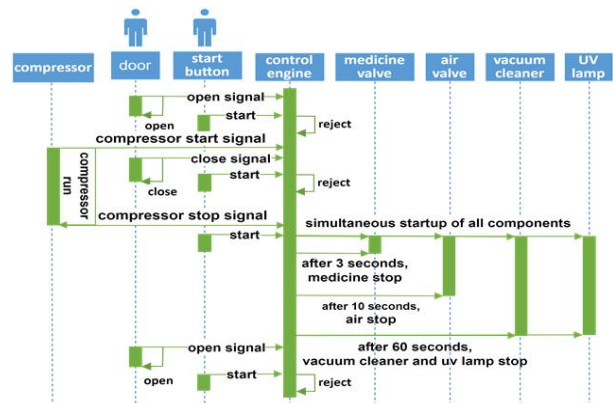


Fig. 3.5. shoes cleaning start while the compressor is running

신발 세척 프로세스가 시작되기 전에 컴프레서가 작동을 시작하고, 그 이후 신발 세척 프로세스를 시작할 경우 시작 신호에 대해 무시하도록 한다. 컴프레서 작동이 멈춘 후 신발세척 프로세스를 시작하면 Fig. 3.3과 같이 정상 상태의 시나리오로 작동한다.

2.4 running the compressor during shoes cleaning

시나리오 2.3과 달리 이미 신발 세척 프로세스가 시작된 이후에 컴프레서의 작동이 시작될 경우 신발 세척 프로세스는 컴프레서 작동에 영향을 받지 않고 정상 상태로 종료한다. 그러나 컴프레서 작동 중에 신발 세척 프로세스가 정상 종료 한 후 새로운 신발을 세팅하고 시작을 하고자 할 때는 컴프레서가 종료될 때 까지 시작을 하지 않는다. 다음의 Fig. 3.6은 신발 세척 프로세스가 진행 중에 컴프레서 작동이 시작된 상황에 대한 시나리오이다.

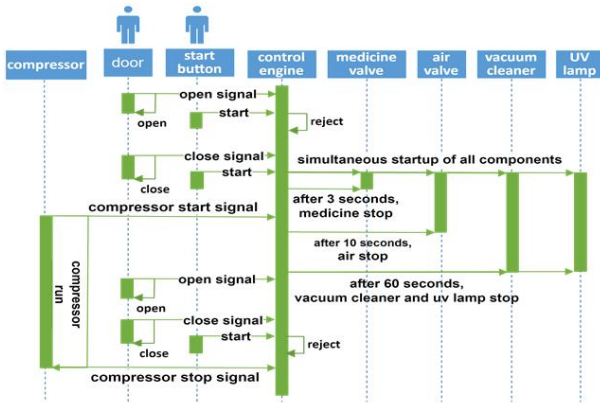


Fig. 3.6. running the compressor during shoes cleaning

Fig. 3.6에서 신발 세척 프로세스 작동 중 컴프레서가 가동되는 것은 신발 세척 프로세스에 영향을 주지 않지만, 컴프레서가 가동 중에 새로운 신발 세척 프로세스는 시작할 수 없음을 확인할 수 있다.

4. shoes cleaner control scenarios summary

연구 도메인인 신발세척기와 신발세척기 작동 시나리오 분석을 통해 얻을 수 있는 조건으로 다음의 Table. 3.2와 같은 구동 장치들을 확인할 수 있다.

Table 3.2. a summary of sensor and actuator for shoes cleaner

	component	explanation	criteria
I N	compressor sensor	compressor run detection	AC 220V detection
	door sensor	door open detection	DC 5V detection
	start switch	shoes cleaning start	DC 5V detection
O U T	LED light	LED to indicate shoes cleaning is in progress	DC 5V
	sanitizer valve	medicine supply ON/OFF	DC 5V
	air valve	air ON/OFF	DC 5V
	vacuum cleaner switch	vacuum cleaner START/STOP	AC 220V
	uv lamp power switch	uv lamp ON/OFF	AC 220V

시나리오 ①~④를 구성하는 상태에는 정지(s:stop), 준비(r:ready), 세척(c:cleaning)이 있다. 정지 상태는 세척을 하지 않으면서 도어가 개방 되어 있거나, 컴프레서가 작동 중일 때 인데, 시작 버튼에 의해 곧바로 세척 상태로 전이될 수 없는 상태이다. 정지 상태에서 세척실 도어가 닫히고, 컴프레서가 작동중이 아니면 준비 상태로 전이된다. 준

비 상태에서 시작 버튼에 의해 세척 상태로 전이되지만, 세척 상태로 전이되기 전에 컴프레서가 작동하면 정지 상태로 복귀한다. 이를 다음의 Table. 3.3에 요약하였다.

Table 3.3. shoes cleaner state transition table

	Start	dOpen	dClose	cRun	cStop	timerOut
s	s	s	r	s	r	∅
r	c	s	∅	s	∅	∅
c	c	s	∅	c	∅	r

event summary Start:push the start button, dOpen/dClose:door open/close, cRun/cStop: compressor run/stop, timerOut:shoes cleaning complete

Table. 3.3에서 세로축은 상태를, 가로축은 이벤트를 나타내고, 세로축의 상태가 가로축의 이벤트 발생에 대응해서 다음 상태로 전이되는 것을 표로 표현한 것이다. Table. 3.3에서 ∅는 무관조건(don't care condition)으로서 이벤트가 상태 전이에 영향을 주지 않음을 의미한다. Table. 3.3의 상태 전이표를 그래프로 표현하면 다음의 Fig. 3.7과 같다.

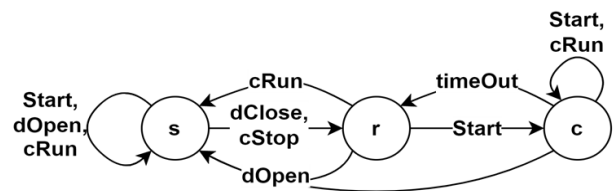


Fig. 3.7. shoes cleaner state transition diagram

Fig. 3.7에서 s->r->c로 전이되는 경로는 시나리오 ①에 의해 발생하는 전이이다. 이벤트 dOpen은 r->s 또는 c->s의 전이를 발생시키며 시나리오 ②에서 설명하는 경로이다. 상태 c에서 회귀하는 이벤트 cRun은 시나리오 ③에서 설명하는 경로이며, 상태 s에서 회귀 이벤트 cRun과 s->r로 가는 경로상의 이벤트 cStop은 시나리오 ④를 나타낸다. Fig. 3.7의 상태 전이도 구현은 다음 4장에서 기술한다.

IV. Implementing shoes cleaner control system

신발 세척기 컨트롤 시스템은 제어 엔진으로 라즈베리 파이를 이용하며, 라즈베리파이에 의해 생성되는 제어 신호를 수행하기 위한 구동 장치들로 구성된다. 구현은 신발 세척기 컨트롤 시스템을 구성하는 하드웨어 구성요소의

요약과 회로도에 대한 사항 및 제어 엔진의 로직 구현에 대한 사항을 기술한다.

1. hardware components for shoes cleaner control system

컨트롤 시스템의 제어 로직을 운영하기 위해 라즈베리파이를 이용하여 제어엔진을 구현한다. 다음 Table. 4.1은 제어 엔진 구현에 적용되는 라즈베리파이의 스펙을 요약한 것이다.

Table. 4.1 control engine device specification

모델	RaspberryPi 2 Model B	
CPU	900Mhz Quad core ARM Cortex-A	
Memory	1GB	
I/O	USB	USB 2.0 4 Port
	GPIO	40Pin GPIO
	Ethernet	100BaseT Ethernet
Multimedia	Display	FULL HDMI Port
	Audio	3.5mm stereo audio jack
rlxk	Storage	MicroSD Slot

라즈베리파이를 이용한 제어엔진은 라즈베리파이에 장착되어 있는 40Pin GPIO를 이용하여 센서 입력과 제어 신호 출력을 수행함으로써 제어로직을 실현한다. 다음의 Table. 4.2는 라즈베리파이로 구현하는 제어엔진의 GPIO에 할당된 제어 신호이다.

Table. 4.2. control engine GPIO pin assignment

GPIO PIN No.	signal	I/O direction	WiringPi Pin No.	roll of pins
1	3.3V	~	~	power
2	5V	~	~	power
7	MEDICINE	OUT	8	medicine discharge control
11	AIR	OUT	7	air discharge control
12	COMP	IN	1	compressor run detection
13	CLEANER	OUT	0	vacuum cleaner start/stop control
15	LED	OUT	2	run indicate LED control
16	DOOR	IN	4	door open/close detection
18	SWITCH	IN	5	start switch push detection
19	SUB_1	OUT	12	reserved control 1
21	SUB_2	OUT	13	reserved control 2

Table. 4.2에서 확인할 수 있는 바와 같이 제어 엔진의 GPIO 핀 할당은 Table. 3.2의 센서 및 구동장치들에 대해서 할당하였으며, 3개의 입력과 6개의 출력으로 할당하였다. 한편 Table. 4.2에서 WiringPi Pin No는 라즈베리

파이에서 GPIO 제어 프로그래밍을 편리하게 하기 위해 지원되는 WiringPi 라이브러리에서 GPIO 핀에 할당된 핀 번호이다. 즉 라즈베리파이 GPIO 제어는 WiringPi 핀 번호를 이용하여 프로그래밍 할 수 있도록 한다.

2. design an I/O board

제어엔진에서 제어로직에 의해 생성되는 제어 신호를 구동장치에 전송하거나 센서로부터 획득된 값을 제어엔진에서 수신하는 GPIO는 동작전압 3.3V, 허용전류 3mA로 규격화 되어 있다[1]. 그렇기 때문에 Table. 3.2에서 정의한 제약과 압축공기 제어를 위한 밸브 및 LED 램프를 구동 할 수 없으며, 각종 센서에서 들어오는 입력값을 안전하게 수신하지 못하는 문제가 있다.

이와 같은 문제를 해결하기 위해 입출력 보드는 전원부(power supply)를 갖추고 있으며, 그 외 제어 엔진 연결부(control engine connection part), 신호 전송부(signal transfer module), 센서 및 구동장치 연결부(I/O connection part)를 구성하고, 전원부에서 생성하는 전원 값을 라즈베리파이에 공급하거나 신호 전송부에 의해 인가되는 구동장치로 공급하는 역할을 수행한다. 제어 엔진 연결부는 라즈베리파이의 GPIO와 Tab 4.2에서 정의한 신호들을 입출력 보드와 연결하는 연결부이며, 11개 신호를 수용할 수 있도록 26핀 리본 커넥터(ribbon connector)를 통해 라즈베리파이와 연결한다.

제어 엔진에 의해 생성되는 제어 신호는 GPIO를 거쳐 입출력보드의 연결부를 통해 신호 전송부로 전달되는데, 신호전송부는 제어 엔진의 제어 신호에 따라 3.3V로 구동되는 릴레이를 작동 시키며, 릴레이는 연결된 GPIO 핀 역할에 따라 5V 또는 3.3V 전압을 구동장치 연결부 또는 제어엔진 연결부로 전송하도록 한다. 다음의 Fig. 4.1은 제어 엔진의 GPIO 핀들이 신호전송부를 통해 각종 구동장치 또는 센서들과의 연결 관계를 나타내는 회로도 이다.

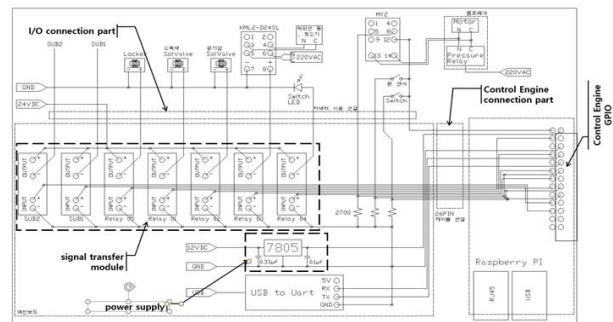


Fig. 4.1. circuit diagram for I/O board

3. implementing I/O board

입출력 보드는 Fig. 4.1의 회로도에 따른 보드를 PCB(printed circuit board)에 실장 한 것으로서 다음의 Fig. 4.2와 같다.

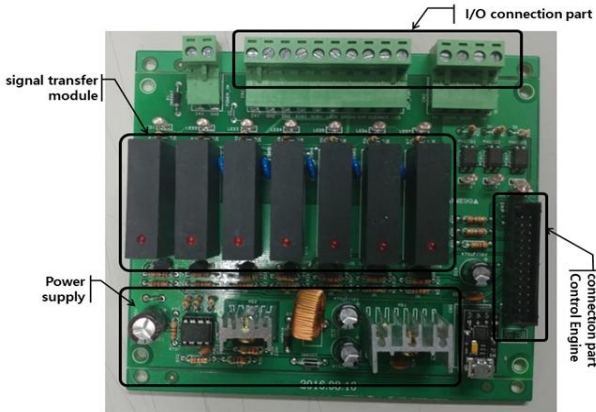


Fig. 4.2. implemented I/O board

Fig. 4.2의 입출력 보드는 Fig. 4.1의 입출력 보드 회로도를 그대로 반영한 보드이며, Fig. 4.1의 주요 구성요소들에 대한 레이블을 Fig. 4.2의 I/O보드에서 동일한 부분에 표시하였다. 한편 다음 Fig. 4.3은 Fig. 4.2의 보드와 제어엔진 그리고 센서 및 구동장치들과의 연결을 개념적으로 표현한 것이다.

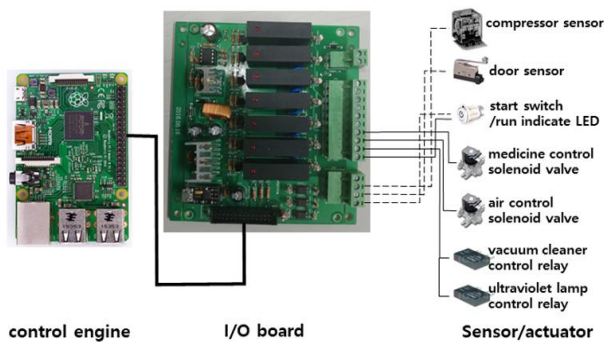


Fig. 4.4. control system conceptual diagram

Fig. 4.4에서 I/O 보드와 센서/구동장치 사이 실선은 출력이고, 점선은 입력을 나타낸다. 그리고 제어 엔진과 I/O보드 사이에는 26채널 리본 커넥터를 이용한 연결은 다음 Fig. 4.5와 같다.

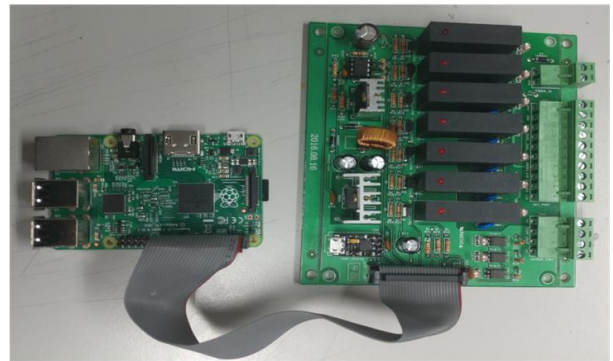


Fig. 4.5. connecting between control engine and I/O board with 26 channel ribbon connector

4. implementing control logic

제어로직은 Fig. 3.7의 상태 전이도를 구현한 것으로서 라스베리파이의 GPIO를 제어하기 위해 wiringPi 라이브러리를 필요로 한다. 그리고 Fig. 3.7 상태 전이도에서 확인할 수 있는 각종 이벤트들의 발생 시점에 실시간으로 발생 사실을 인지하기 위해 폴링(polling)방법을 적용하며, 스레드(thread) 프로그래밍을 해야 하므로 스레드 라이브러리도 필요하다. 이를 반영한 헤더 파일 선언부는 다음과 같다.

```
...
#include <wiringPi.h>
#include <pthread.h>
```

헤더파일 선언부에서 ‘...’으로 표기한 부분은 c 표준 라이브러리용 헤더 파일들이서 생략하였다. 한편 Fig. 3.7 신발세척기 상태 전이도를 구현하기 위해서는 Table. 4.2의 GPIO 핀 할당에서 wiringPi 핀 번호를 선언하면 다음과 같다.

```
#define MEDICINE      8 //out - 약액 밸브 제어
#define AIR           7 //out - 압축공기 밸브 제어
#define VACUUM       0 //out - 진공청소기(uv 램프)
#define SWITCH_LED   2 //out - 스위치 LED

#define COMPRESSOR   1 //in  컴프레사 센서
#define DOOR         4 //in  도어 개폐 센싱
#define SWITCH       5 //in  시작 스위치 센싱
```

한편 Table. 3.1의 작동 조건을 지정하기 위한 시간 변수들과 제어로직 구현에서 활용할 태그 변수들 그리고 신발 세척을 위한 쓰레드 함수 및 변수를 다음과 같이 선언한다.

```
// tag 변수
bool ButtonTag = false; // 시작/정지 버튼 값 상태
bool MachineTag = false; // 세척 시작/진행 중 표시
bool DoorTag = false; // 도어 열림/닫힘 상태
bool AirTag = false; // 컴프레서 작동/중지 표시

// 제어 조건을 위한 시간변수
// 각 시간 변수의 단위 시간 : 100ms
int AirTime = 100; // 압축 공기 토출 시간 10초
int MediTime = 30; // 약품 토출 시간 3초
int ProcessTime = 600; // 신발 세척 시간 60초

// cleaningThread를 위한 thread 함수 및 변수
void *cleaningThread(void * _Get);
pthread_t t_Cleaning;
```

이상과 같은 상태에서 프로그램의 시작과 동시에 GPIO에 대한 설정 및 초기화 작업을 수행해야 하는데, 이를 반영한 코드는 다음과 같다.

```
if (wiringPiSetup()==-1) { // wiringPi setup
    printf("Fail "); return 0;
}

// 구동장치를 위한 GPIO 설정
pinMode(AIR, OUTPUT);
pinMode(MEDICINE, OUTPUT);
pinMode(SWITCH_LED, OUTPUT);
pinMode(VACUUM, OUTPUT);

// 센서 값 입력을 위한 GPIO 설정
pinMode(COMPRESSOR, INPUT);
pinMode(SWITCH, INPUT);
pinMode(DOOR, INPUT);

// 모든 출력 핀에 ON 시그널 전송
digitalWrite(MEDICINE, ON);
digitalWrite(AIR, ON);
digitalWrite(VACUUM, ON);
digitalWrite(SWITCH_LRD, ON);

// 3초간 유지
sleep(3000);

// 모든 출력 핀에 OFF 시그널 전송
digitalWrite(MEDICINE, OFF);
digitalWrite(AIR, OFF);
digitalWrite(VACUUM, OFF);
digitalWrite(SWITCH_LRD, OFF);
```

이제 제어엔진 구동에 필요한 초기 조건이 완성되었으므로 제어 엔진을 구동 해야 한다. 제어 엔진의 구동은 신발 세척기의 여러 상태들을 모니터링 하는 것 에서부터 시작해야 한다. 모니터링 모듈은 폴링 기법을 통해 구현하였으며, 시스템이 종료할 때 까지 무한 루프 형태로 작동한다. 다음의 Fig. 4.6은 이벤트 모니터링 모듈의 모니터링 절차를 플로우차트로 추상화 한 것이다.

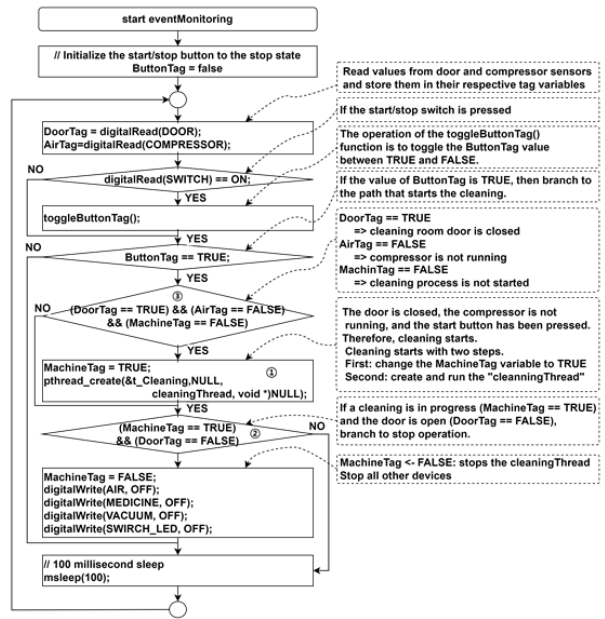


Fig. 4.6. shoes cleaner eventMonitoring module

eventMonitoring 모듈은 무한루프 상태에서 100ms 단위로 한 주기를 모니터링 한다. 모듈에서 ① 위치의 코드가 시나리오 3.2.1을 수행하는 코드이다. GPIO의 SWITCH핀에 대한 입력이 감지되었을 때부터 시작하여 모든 조건을 검사하고, 모두 정상 상태일 때 세척을 시작한다. 세척의 시작은 cleanningThread를 생성하고 실행함으로써 세척을 실행한다. ②위치의 시나리오 3.2.2를 실행하기 위한 조건 검사이며, 현재 세척을 실행 중인 상태에서(MachineTag == TRUE) 세척실 도어가 열리면(DoorTag == FALSE) 곧바로 모든 실행을 중단 시킨다. 이 때 MachineTag 변수는 eventMonitoring 모듈과 cleanningThread 사이에 제어변수로 공유되기 때문에 MachineTag 값을 FALSE로 설정하는 순간 cleanningThread는 세척 과정을 중단하고 스레드를 종료한다. ③은 시나리오 3.2.3을 검사하는 조건문으로서 AirTag의 값이 TRUE이면 100ms 동안 sleep하는 코드로 이동하여 다음 사이클로 이동하도록 하며, 다음번 주기에서 다른 조건을 평가할 수 있도록 한다. 시나리오 3.2.4 역시 ③의 MachineTag 값을 이용하여 평가 되는데, MachinmTag 값이 TRUE일 경우 시나리오 3과 동일한 경로를 통해 진행되기 때문에 ArTag 값의 상태와 무관하게 진행된다. 다음의 Fig. 4.7은 ①의 위치에서 생성하는 cleanningThread의 스레드 함수를 플로우 차트로 추상화 한 것이다.

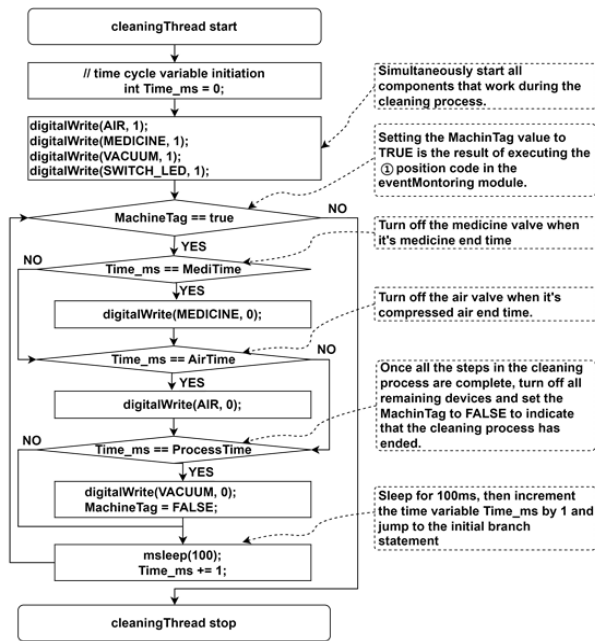


Fig. 4.7. shoes cleaner cleaningThread module

Fig. 4.7에서 가장 우선적으로 시간 측정 변수 Time_ms 변수를 초기화 한 후 세척에 참여하는 모든 장치들(압축공기, 약액, 진공 청소기, UV 램프, 스위치 LED)를 동시에 가동 시킨다음 MachineTag 값을 검사한다. MachineTag 값은 Fig. 4.6 eventMonitoring 모듈의 코드 ①에서 TRUE로 설정하고 cleaningThread를 생성하였기 때문에 첫 번째 검사에서는 당연히 TRUE로 평가된다. 그런 다음 반복문의 반복 주기에 따라 약품, 압축공기, 전체 세척 과정으로 설정된 시간 값에 따라 차례대로 종료한다. 모든 세척 과정을 완료 하였을 경우 MachineTag 값을 FALSE로 수정 함으로서 스레드를 종료 할 수 있도록 한다.

Fig. 4.7이 정상적으로 수행되는 상황이어도 eventMonitoring 모듈의 ② 또는 ③의 위치에서 도어 열림 또는 기타의 이벤트를 감지 하였을 경우 eventMonitoring 모듈에서 MachineTag 변수를 FALSE로 설정할 수 있으며, 이 경우에도 즉각적으로 세척 스레드를 종료하게 된다.

V. Evaluation for implementing system

이 논문에서 기술한 신발 세척기 제어 시스템은 오픈 소스 하드웨어 플랫폼 중에서 가장 대표적 SBC인 라즈베리 파이를 제어 엔진으로 하고, I/O보드를 통해 각종 밸브 및 장치들을 제어 시나리오에 따라 자동으로 제어할 수 있는 시스템이다.

개발 시스템에 대한 평가는 시범 설치된 신발 세척기1 세트에 대하여 2023년 6월 ~ 2024년 5월, 1년 동안 운영하는 과정 분석을 통해 개발 시스템의 특성을 분석하는 것과 개발 시스템을 통해 세척한 신발의 세척 결과에 대한 분석으로 이루어진다.

1. Analyze the pilot system

개발 시스템 한 세트를 협력기관에 설치하고, 운영 과정에서 발생한 여러 사항 들을 데이터화하여 분석한다. 협력 기관은 직원 수 150여명이 근무하는 기관이며, 작동 과정에서 발생하는 소음 문제로 실외 흡연구역 옆 공터에 비가림 시설을 설치한 후 신발 세척기를 설치하여 접근성을 최대로 확보할 수 있도록 하였다. 다음의 소 절에서 운영 결과를 요약한다.

1.1 Medicine usage

측정 대상 중에서 약액의 경우 세척기 후면부에 150ml의 통을 충전하여 공급하는데, 1회 사용 약액이 평균 0.2ml 정도가 투입되는 점을 감안하면 1회 충전으로 대략 750쪽 정도의 신발을 세척하는 것으로 계산 할 수 있다. 다음의 Fig. 5.1은 월별로 약액 사용량에 대해 요약한 것으로서 월별로 충전한 약액 량을 ml 단위로 측정하여 반영한 내용이다.

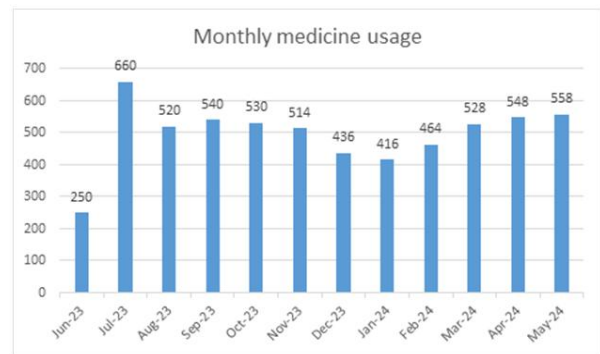


Fig. 5.1. monthly medicine usage

Fig. 5.1을 통해 확인할 수 있는 사항은 계절별로 사용량의 변화가 있음을 확인할 수 있으며, 2023년 6월의 경우 설치한 첫 달이어서 사용량이 저조한 것으로 보이며, 7월의 경우 급격히 늘어난다. 8월의 경우 휴가기간이 있고, 9월의 경우 추석 연휴가 있어서 사용량 조정이 있는 것으로 보이며, 겨울에 상대적으로 사용량이 줄어드는 것을 확인할 수 있다

1.2 Content of the repair request

1년 동안의 운영 과정에서 수리 요청 접수 3건이 있었으며, 이를 다음의 Table 5.1에 요약 하였다.

Table 5.1. Summary of the repair request

reception number	reception month	fault part	actions to take
1	Nov. 23	vacuum cleaner	changing dust bags
2	Feb. 24	air compressor	Compressor air cleaner filter changing
3	May. 24	start/stop switch	Switch contact fault

Table 5.1에서 확인할 수 있는 수리요청 내용 중에서 1번 및 2번은 진공청소기와 컴프레서의 일상적 점검 항목에 해당하는 것이어서 기능적으로 문제가 되지 않으나, 3번의 경우 시작/정지 버튼의 접점 불량이어서 기능적 문제가 발생하는 사항이다. 이 문제가 발생한 원인은 세척실 내부에 노출되어 있는 시작/정지 버튼의 접점 부분이 신발 세척 과정에서 탈거된 이물질의 유입으로 접점 불량을 발생시키는 것으로 분석되었으며, 이에 대응하여 시작/정지 스위치의 접점 부분은 글루건을 이용하여 밀폐함으로써 간단하게 해결할 수 있는 문제였다.

1.3 Operational environments

실외에 설치하면서 눈/비를 가릴 수 있는 시설 설치가 함께 이루어졌으며, 그 결과 우천시에도 사용하는데 큰 어려움이 없도록 설치하였다. 그러나 실외에 설치한 사항이어서 온도 및 습도 등의 영향을 받을 수 있는 여지가 매우 큰 상황이었다. 사용 환경을 측정하기 위해 라즈베리파이에 온도 센서를 연결하여 측정한 온도 값을 월별로 요약하면 다음의 Fig. 5.2와 같다.

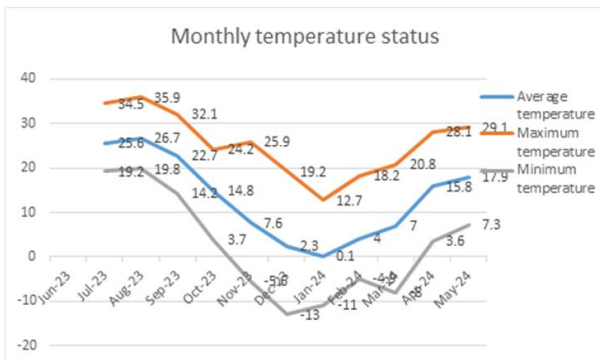


Fig. 5.2. Summary of monthly temperature status

Fig. 5.2에서 최고 온도는 35.9도이고, 최저 온도가 -13도에 달하는 상황에서 앞의 ②소절에서 다른 오류 사항 외에 특별한 오류 없이 운영하였으므로 매우 견고한 시스템이라 평가 할 수 있다.

2. Analyze shoes cleaning effectiveness

이 논문에서 기술한 실발세척기의 세척 효과 분석을 위해 Table 5.2와 같이 시료를 준비하였다.

Table 5.2. sample for test and evaluation

kinds of shoes	male	female	sum
safety shoes	3	2	5
golf shoes	0	5	5
low shoes	5	0	5
regular shoes	5	0	5

Table 5.2의 시료에 대하여 해당 시료를 착용해야 하는 업종의 근로자들을 대상으로 시료를 지급하고 4주 동안 착용하게 하면서 착용 기간동안 수시로 잘 착용하고 있는지를 확인하였다. 4주간의 착용 기간 경과 후 일괄 수거하여 한국의류시험연구원에 의뢰하여 시험 평가를 시행하였으며, 평가 항목 및 방법은 다음과 같다.

- ① 살균력 : 식품의약품안전청의 국가표준시험방법인 세균현탁액시험방법 적용
- ② 소취성(냄새제거) : 일본섬유평가기술협회 소취시험 합격기준 적용
- ③ 세척효율(이물질 제거 - 중량감소율) : 세척 후 측정 중량/세척 전 측정 중량 × 100
- ④ 세척시간 : 시작버튼을 누르면서 측정을 시작하여 세척 완료와 함께 측정 종료

이 중에서 ③항의 경우 측정값의 객관성을 담보하기 위해 공인시험기관에서 보유한 실험용 정밀 전자저울을 이용하여 측정하였다. 다음의 Table 5.3은 시료별 시험평가 결과를 요약한 것이다.

Table 5.3. result of test and evaluation

sample No.	method ① (%)	method ② (%)	method ③ (%)	method④ (second)
1	99.9	82	99.95	61
2	99.9	88	99.96	60
3	99.9	86	99.96	61
4	99.9	84	99.97	60
5	99.9	84	99.96	60
6	99.9	86	99.95	60
7	99.9	84	99.97	60
8	99.9	88	99.94	60
9	99.9	82	99.96	60
10	99.9	86	99.97	61
11	99.9	82	99.94	60
12	99.9	88	99.95	60
13	99.9	84	99.97	60
14	99.9	86	99.96	60
15	99.9	88	99.95	60
16	99.9	84	99.95	61
17	99.9	86	99.97	61
18	99.9	86	99.97	60
19	99.9	88	99.97	60
20	99.9	86	99.95	60
average	99.9	85.4	99.9585	60.25

Table. 5.3에서 method ①인 살균력은 모든 시료에도 편차 없이 99.9%를 나타내고 있다. 이는 신발 내부 위생에 중요하게 작용할 수 있음을 나타낸다. method ② 소취성은 냄새의 원인이 되는 암모니아 농도 측정을 통해 얻을 수 있는 값으로서 신발 내부에 이물질로 부착되어 있는 각 질에 의해 발생하는 악취를 제거한 결과이다. method ② 역시 상당한 성과를 얻은 것으로 확인할 수 있다. method ③ 세척 효율은 세척 후 중량을 세척 전 중량으로 나누어 얻는 값으로서 세척 과정에서 신발 내부 이물질 제거에 의해 발생하는 중량차를 측정하는 것이다. 중량 측정의 정밀도 확보를 위해 시험평가기관에서 보유하고 있는 정밀 전자저울을 이용하여 측정하였으며, 그 결과 평균 0.04% 정도의 중량 감소를 확인할 수 있다. 0.04%의 감소율이 미미하다고 할 수도 있으나, 신발 내부 이물질의 형태와 중량을 감안할 때 신발 내부 이물질 제거가 확실히 발생했음을 확인할 수 있다. 마지막으로 method ④ 세척시간은 시작 버튼이 눌린 다음 종료할 때 까지의 시간을 측정하는 것으로서 제어 엔진의 로직에 따라 60초로 귀결되어야 하겠지만, 측정 시계를 제어엔진과 연동하여 측정하는 것이 아니라 시험평가기관의 연구원이 타이머 가동을 통해 측정하다보니 발생한 오차로 판단할 수 있으며, 60초로 수렴되는 시간동안의 세척을 통해 method ① ~ method ③의 결과를 도출하고 있고, 각 시료들 사이의 편차가 크지 않게 나타나고 있음을 볼 때 제어 엔진에 의한 제어가 정확하게 작용하고 있음을 확인할 수 있다.

VI. Conclusion and future work

이 논문에서는 물을 사용하지 않으면서, 압축공기와 약액 및 자외선램프를 이용하여 신발 내·외부를 세척 하고 살균·소독하는 신발세척기의 제어 시스템을 개발하였다. 신발 세척기는 실외에 설치하였음에도 불구하고 1년의 운영기간 동안 환경적 요인에도 불구하고 고장 없이 잘 작동 할만큼 견고한 시스템임을 확인할 수 있다. 아울러 제어 시스템의 제어에 의해 목표로 하는 세척 성능을 발휘하도록 작동하였으며, 그 결과 우수한 신발 세척 결과도 얻을 수 있었다.

이 논문에서 개발한 신발세척기 제어 시스템은 제어엔진으로 라즈베리파이를 이용하였으며, 신발 세척 과정에서 발생하는 각종 이벤트를 모니터링 하기 위해 폴링 방법을 적용하였다. 그리고 폴링(polling) 모듈과 병행(parallel) 수행되면서 신발 세척 과정을 제어하는 모듈은 스레드를 이용하여 구현하였다. 그러나 폴링 방법은 전력 등 자원 소모가 많기 때문에 인터럽트(interrupt)를 이용한 방법으로 개선할 필요가 있다. 아울러 외부와의 통신 기능을 구현하지 않고 있는데, 블루투스(bluetooth)등을 이용한 외부 장치와의 통신을 통해 신발 세척기의 설정값을 변경할 수 있도록 하거나 또는 신발 세척기 내부 세척 관련 로그 데이터 수집을 통해 세척 패턴 분석 등을 할 수 있도록 개선할 필요가 있다.

REFERENCES

- [1] Sung Woo Kim, "RaspberryPi with Internet of Things(Revised)", JPub, pp. 3~4, 2022
- [2] Singh, D., Sandhu, A., Thakur, A. and Priyank, N. "An Overview of IoT Hardware Development Platforms". International Journal on Emerging Technologies, 11(5): 155-163. 2020
- [3] Kyungyong Heo , "Arduino with the Internet of Things", JPub, pp.32~33, 2018
- [4] "ESP32-C6 Series Datasheet", Esspresif, 2024, https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf
- [5] Triwiyanto Triwiyanto et.al. , "Embedded machine learning using a multi-thread algorithm on a RaspberryPi platform to improve prosthetic hand performance", Micromachines 2022, 13(2), pp. 191, 2022, DOI: <https://doi.org/10.3390/mi13020191>
- [6] Ashwin Pajankar, "Raspberry Pi supercomputing and scientific programming", Apress Berkeley, CA, 2017, DOI: <https://doi.org/10.1007/978-1-4842-2878-4>

- [7] S.Alex David, S.Ravikumar, A. Rizwana Parveen, "Raspberry Pi in computer science and engineering education", Intelligent Embedded Systems. Lecture Notes in Electrical Engineering, vol 492., pp. 11~16, Springer, 2018, DOI:https://doi.org/10.1007/978-981-10-8575-8_2
- [8] G.Senthilkumar, K.Gopalakrishnan , V. Sathish Kumar, "embedded image capturing system using raspberry pi system", International Journal of Emerging Trends & Technology in Computer Science, Vol. 3, Issue 2, pp. 213~215, 2014,
- [9] Samarth Brahmabhatt , "Embedded computer vision: Running OpenCV programs on the Raspberry Pi in Pratical OpenCV", Apress, pp.201~218, 2013, DOI:https://doi.org/10.1007/978-1-4302-6080-6_11
- [10] Mochammad Ariyanto, et. al. , "Real-time image processing method using Raspberry Pi for a car model", 2019 6th International Conference on Electric Vehicular Technology (ICEVT), Bali, Indonesia, 2019, pp. 46-51, DOI: 10.1109/ICEVT48285.2019.8993866.
- [11] Gianluca Cornetta, Abdellah Touhafi, "Design and evaluation of a new machine learning framework for IoT and Embedded devices", Electronics 2021, 10(5), pp. 600, DOI: <https://doi.org/10.3390/electronics10050600>
- [12] Donald Norris, "Machine learning with the Raspberry Pi - Experiment with data and computer vision", Apress Berkeley, CA, 2020, DOI: <https://doi.org/10.1007/978-1-4842-5174-4>
- [13] Delia Velasco-Montero et. al. "Performance analysis of real-time DNN inference on Raspberry Pi," Proc. SPIE 10670, Real-Time Image and Video Processing 2018, 106700F (14 May 2018); DOI: <https://doi.org/10.1117/12.2309763>
- [14] Achraf Jmal, et. al., "An embedded ANN Raspberry Pi for inertial sensor based human activity recognition", The Impact of Digital Technologies on Public Health in Developed and Developing Countries. ICOST 2020. Lecture Notes in Computer Science, vol 12157. pp.375~385, Springer, DOI:https://doi.org/10.1007/978-3-030-51517-1_34
- [15] M. Vanitha, M. Selvalakshmi and R. Selvarasu, "Monitoring and controlling of mobile robot via internet through raspberry Pi board," 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM), Chennai, India, 2016, pp. 462-466, DOI: 10.1109/ICONSTEM.2016.7560864.
- [16] G. Vieira, J. Barbosa, P. Leitão and L. Sakurada, "Low-Cost Industrial Controller based on the Raspberry Pi Platform," 2020 IEEE International Conference on Industrial Technology (ICIT), Buenos Aires, Argentina, 2020, pp. 292-297, DOI: 10.1109/ICIT45562.2020.9067148.

Authors



Deukchang Hyun received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Kwangwoon University, Korea, in 1990, 1992 and 1997, respectively. Dr. Hyun joined the faculty of the Department of Information and

Communication Engineering at Far East University, Eumsung, Korea, in 1998. He is currently a Professor in the Department of AI Computer Engineering, Far East University. He is interested in internet and mobile computing, and AI computing.