

DC 모터 드라이버를 이용한 라즈베리 파이 기반의 장난감 자동차 제어 시스템

이혜성 · 이창우

국립 군산대학교

Raspberry Pi-based Toy Car Control System Using DC Motor Driver

Hye Seong Lee · Chang Woo Lee

Kunsan National University

E-mail : ddaafftt@kunsan.ac.kr, leecw@kunsan.ac.kr

요 약

IoT(사물 인터넷)는 더 이상 개념에만 머물러있지 않고, 개인 사용자가 직접 소형 보드와 부품을 통해 구현을 해보는 등 일상화되었다. 본 연구는 일반 장난감 자동차를 라즈베리 파이와 모터 드라이버를 이용하여 제어하는 시스템을 구현하였다. 이 연구를 통해 일반 장난감 자동차를 RC 카로 만들거나 추가적인 개발로 학습을 하고 자율주행을 하는 등의 발전을 시킬수 있을 것이다.

ABSTRACT

IoT(Internet of Things) is no longer just a concept, and It has become common as individual users have tried to implement it directly through small boards and components. This research implemented a system to control a general toy car using Raspberry pi and motor driver. This research will help develop ordinary toy cars to RC cars or learn on their own and drive on their own through further development.

키워드

Raspberry Pi(라즈베리 파이), Toy Car(장난감 자동차), Python(파이썬)

I. 서 론

미국의 정보 기술 자문 및 시장 조사 기관인 Gartner는 잠재성이 있는 기술을 선정하여 발표를 하는데 사물 인터넷 분야에 특화된 2017·2018년 10대 상위 기술을 따로 발표하기도 하였고[1], 언론이나 기타 매체에서도 사물 인터넷이 언급될걸 찾아 볼 수 있는 등 IoT는 이제 상당히 일상화된 개념이 되었다. IoT가 빠르게 정착된 건 잘 조성된 인프라로 인해 어느곳에서든 인터넷에 접속을 할 수 있는 것도 있지만 Raspberry Pi나 Arduino와 같은 소형 보드가 보급된 이유도 크다. 이와 같은 보드는 상대적으로 저렴하고 진입이 쉬워서 개인 사용자들도 구매를 하여 직접 구현을 하는 경우도 찾아 볼 수 있는데, 보드에 최적화된 부품을 구매함으로써 추가적인 비용이 들게 된다. 본 연구에서는 비용 절감과 더불어 차별화를 위해 추가적인 부품의 사용을 지양하고 기존의 장난감 자동차를 기반으로한 시스템을 제안한다.

II. 본 론

하드웨어 환경

본 시스템은 Raspberry Pi Model 3 B 보드를 기반으로 구성하였고, 사용된 장난감 자동차의 구성에 따라 3개의 L298N 모터 드라이버가 사용되었다. 전원 공급은 장난감 자동차에 포함되어있던 6V * 2 배터리로 하였다. 그리고 전원 공급 및 Raspberry Pi와 모터 드라이버간의 연결을 위해 브레드보드와 점퍼 케이블도 사용 되었다.

소프트웨어 환경

Raspberry Pi의 운영체제는 Raspberry Pi Foundation에서 배포하는 것 중 2018년 03월 13일에 배포된 Raspbian Stretch를 사용하였다. 프로그램은 Python3으로 작성되었고, 모터드라이버 연동 및 기능 구현을 위해 time과 RPi.GPIO 모듈이 사용되었다.

시스템 설계

장난감 자동차의 경우 후륜을 기반으로 한 이

륜 구동 방식이었고, 후륜의 경우 후륜 하나마다 모터가 하나씩 연결 되어 있었고, 전륜의 경우 모터 하나에 프레임이 연결되어있고, 그 프레임의 움직임으로 두 개의 전륜의 방향을 전환하는 방식이다. 모터가 총 3개이므로 모터드라이버도 그에 맞추어 3개를 사용하였다.

L298N 모터드라이버는 IN1, IN2, IN3, IN4의 핀으로 모터를 제어 할 수 있고 ENA, ENB 핀을 이용하여 PWM 신호를 보낼 수 있다.[2] 이로 인하여 주파수와 Duty-Cycle을 변경할수 있었고, 결과적으로 장난감 자동차의 이동 속도를 제어할 수 있었다.

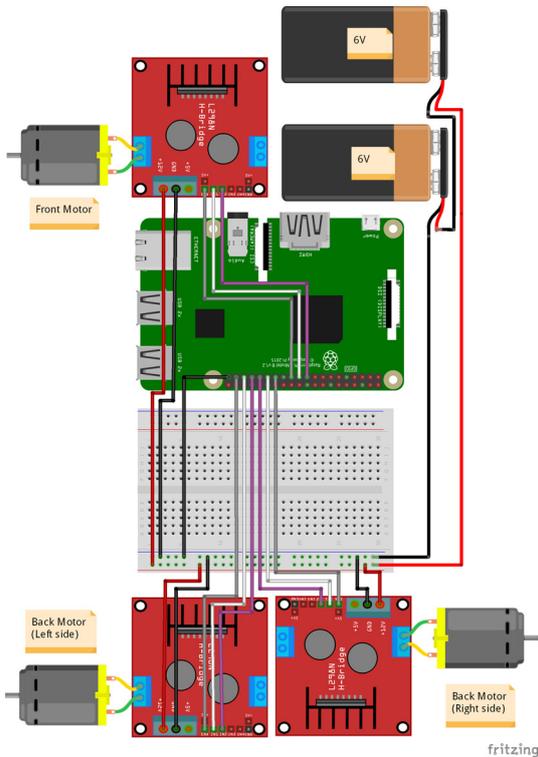


Fig 1. 회로도

GPIO를 통하여 모터를 제어하면 별도의 추가 명령을 내리지 않으면 가장 최근에 받은 명령을 지속적으로 수행하게 한다. 자동차의 전진과 후진은 원하는 시간에 멈춤 명령을 보내면 되지만 이러한 특성 때문에 좌회전이나 우회전과 같은 명령은 원하는 만큼의 방향을 전환하기가 어렵게 된다. 먼저 자동차의 전진과 후진은 후륜을 담당하는 모터들을 동시에 정방향 / 역방향 명령을 수행시키도록 함으로써 구현하였다. 그리고 Duty-Cycle을 변경하지 않는다고 가정하면 원하는 만큼의 방향까지 모터가 도는데 걸리는 시간은 일정하기 때문에 그 시간들을 측정을 하였고, 전륜 담당 모터에게 정방향 / 역방향 명령을 수행시킨뒤 time 모듈의 sleep 함수를 이용하여 원하는 시간만큼 프로그램은 멈추지만 명령은 멈

추지 않고 수행을 하게되고, 프로그램의 정지가 끝나면 명령을 종료 시키는 방안으로 자동차의 방향 전환을 구현하였다.

회로의 구성을 Fig 1과 같이 하였고, 실제로 구성된 모습은 Fig 2에서 볼 수 있다.

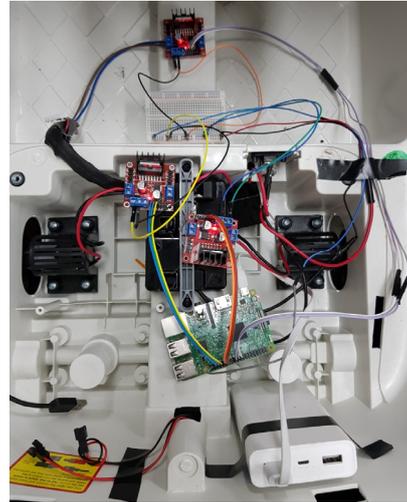


Fig 2. 구현된 시스템

III. 결 론

본 연구에서는 Raspberry Pi나 Arduino와 같은 소형 보드에 최적화된 부품을 사용하지 않고, 기존에 있는 장난감 자동차를 이용하여 시스템을 구성하였다. 추가적인 개발을 통해 이식성을 높인다면 다른 장난감 자동차에도 최소한의 수정만으로 이식을 할 수 있을 것이다. 또한 시중의 컨트롤러나 패드와 호환 가능하게 하면 기존의 자동차를 RC 카로도 제작을 할 수 있고, 카메라를 달아서 영상 전송 뿐만이 아니라 학습을 통하여 자율주행을 탑재하는 등으로도 발전할 수 있는 가능성이 있다.

참고문헌

- [1] <http://www.ndsl.kr/ndsl/issueNdsl/detail.do?tec-hSq=351>
- [2] https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf