창업연계공학설계입문

|  |  |
| --- | --- |
| 프로젝트 명 | Raspberry Car |
| 팀 명 | Leave School |
| 문서 제목 | 4차 과제 보고서 |

|  |  |
| --- | --- |
| Version | 1.0 |
| Date | 2017-11-18 |

|  |  |
| --- | --- |
| 팀원 | 최 주원 / 20\*\*\*\*\*\* |
| 최 호경 / 20\*\*\*\*\*\* |
| 한 정택 / 20171718 |
| 지도교수 | 최 진우 교수 |
| 분반 | 06 분반 |

|  |
| --- |
| CONFIDENTIALITY/SECURITY WARNING  이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 공학설계입문 수강 학생 중 프로젝트 “Raspberry Car”을 수행하는 팀 “Leave School”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “Leave School”의 팀원들의 서면 허락없이 사용되거나, 재 가공 될 수 없습니다. |

문서 정보 / 수정 내역

|  |  |
| --- | --- |
| Filename | 4차 과제 – Raspberry Car.doc |
| 원안작성자 | 최주원, 한정택, 최호경 |
| 수정작업자 | 최주원, 한정택, 최호경 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정 날짜 | 대표  수정자 | Revision | 추가/수정  항목 | 내 용 |
| 2017-11-18 | 한 정택 | 1.0 | 최초 작성 | 최초 작성 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목 차

[1 서론 4](#_Toc497924334)

[2 기본 아이디어 5](#_Toc497924335)

[2.1 H/W 디자인 방향 5](#_Toc497924336)

[2.2 S/W 디자인 방향 5](#_Toc497924337)

[3 수행 내용 6](#_Toc497924338)

[3.1 직교형 장애물 인식을 이용한 직각 코스 주행 6](#_Toc497924339)

[3.2 장애물 인식 및 회피 6](#_Toc497924340)

[3.3 곡률 코스 처리 6](#_Toc497924341)

[4 향후 추진계획 7](#_Toc497924342)

[4.1 향후 계획의 세부 내용 7](#_Toc497924343)

[5 애로 및 건의사항 8](#_Toc497924344)

[6 회의록 9](#_Toc497924345)

# 서론

1. 세 바퀴 구동체 제작

* DC모터(x2), 볼 캐스터, 초음파 센서, 5방향 트랙킹 센서를 이용한 동작 수행
* 전 / 후진이 가능해야 한다.
  + 바닥의 재질에 따른 마찰력, 구동체의 질량, 배터리의 충전 상태를 고려한다
* 초음파 센서를 통한 장애물과의 거리 감지가 가능해야 한다.
  + 온도에 따른 소리의 속도, 장애물의 매질을 고려한다.
    - 소리의 속도 = 약 340m/s (섭씨 15도)
    - 물체 사이의 거리
* 5방향 트랙킹 센서를 통한 흑 / 백색의 구분이 가능해야 한다.
  + 트랙의 재질에 따른 적외선 감지를 고려한다.
    - 반사가 잘되는 검정색 재질 (예 : 투명 테이프가 붙은 라인)

1. 라인 트레이싱 프로그래밍

* 세 바퀴 구동체를 제어하여 트랙을 주행하는 동작 수행
* 직진 코스에서 구동체가 방향을 보정해야 한다.
  + 라인의 굵기와 벗어난 정도에 따른 좌/우 모터의 Duty Cycle을 조정한다.
* 구동체가 곡선 코스를 감지하고 옳은 방향을 판단해야 한다.
  + 5방향 트랙킹 센서가 라인을 감지하는 경우의 수를 고려한다.

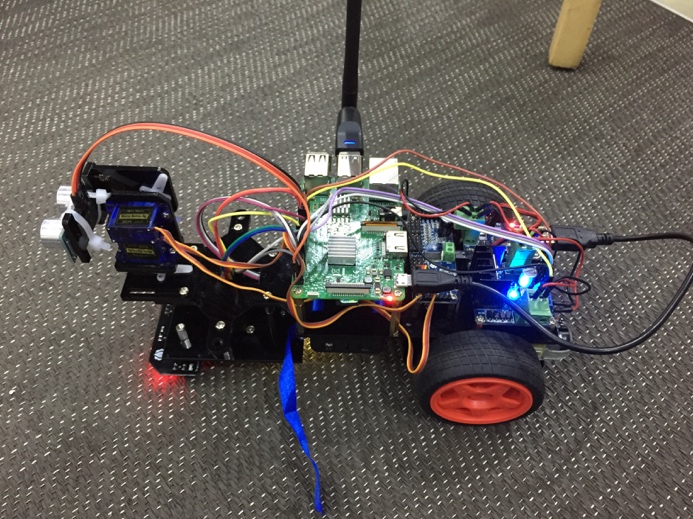
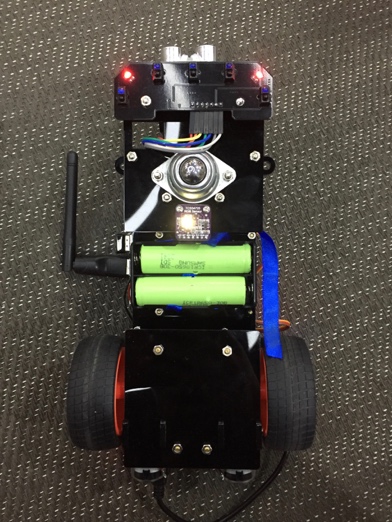
1. 장애물 회피 프로그래밍

* 장애물을 만난 상황에서의 세 바퀴 구동체 제어
* 구동체가 장애물을 만나면 이를 감지하고 회피하는 동작을 수행해야 한다.
  + 구동체의 관성과 초음파 센서의 감지 간격을 고려한다.
  + 구동체가 의도한 정도의 회전을 하도록 조정한다.

# 기본 아이디어

1. 세 바퀴 구동체(H/W) 와 이를 제어하기위해서 이용되는 파이썬(S/W)이 원활한 소통을 할 수 있도록 GPIO포트의 기준을 세운다.
2. 프로젝트를 수행함에 있어서 결과에 영향을 줄 수 있는 변수들(예: 배터리의 충전 상태)을 상쇄해가는 방향으로 작업한다.
3. 장애물을 피하는 동작을 수행할 때 배운 포인트 턴의 개념을 이용한다.
4. 라인트레이싱을 할 때, 5방향 트랙킹 센서가 감지할 수 있는 경우를 생각해보고 각 경우에 대하여 구동체가 대응할 수 있는 모터의 Duty Cycle을 찾아본다.

## H/W 디자인 방향

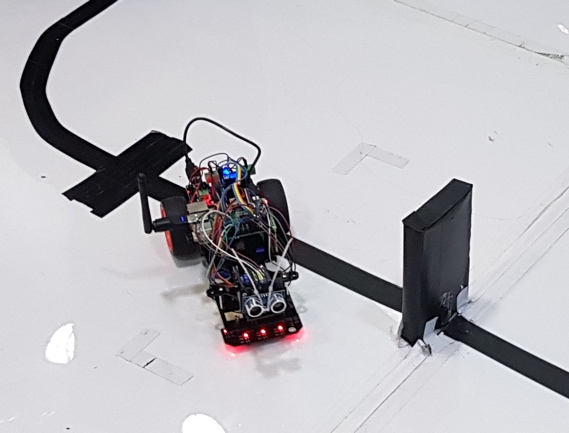
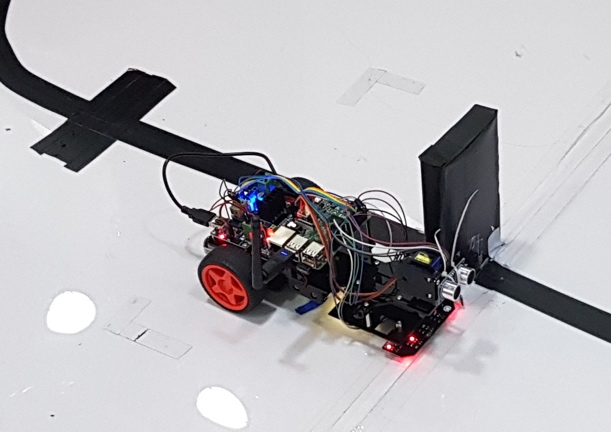
[구동체의 전체적 모습] [사진: 구동체의 하판]

방향에 영향을 주는 볼 캐스터와 DC모터의 위치가 올바르지 않은 위치에 있다면 S/W적으로 보완해야 하는 변수가 더 많아질 수 있다. 따라서 곧게 전/후진 동작을 수행할 수 있도록 위치를 조정해주었다.

초음파 센서가 바라보는 방향이 장애물과 정면으로 마주보는 방향과 다르다면 장애물과 구동체 사이의 거리를 계산함에 있어서 오차가 발생할 수 있다. 따라서 장애물과 일직선으로 바라보도록 초음파 센서의 방향을 조정해주었다.

5방향 트랙킹센서가 바닥에 닿게 된다면 라인을 구분하는데 있어서 문제가 발생할 수 있다. 센서가 바닥에 닿지 않도록 적합한 크기의 Pilar을 사용하여 구동체를 조립했다.

## S/W 디자인 방향

구동체는 크게 직진 코스와 곡선 코스 그리고 장애물을 마주하게 된다. 5 방향 트랙킹 센서가 감지하는 경우에 따라 양쪽 모터에 적절한 Duty Cycle값을 넣어 방향을 조정하게 되는데 센서의 좌우 끝부분이 감지된다면 모터에 값의 차이를 크게 주어 방향을 크게 조정하고 중앙의 좌우가 감지된다면 모터에 값의 차이를 작게 주어 방향을 약하게 조정해주게 된다. 만약 구동체가 장애물을 감지하게 된다면 구동체를 “-]” 모양으로 회전시켜 장애물을 회피한다.

# 수행 내용

라인트레이싱 트랙을 돌고 장애물을 회피하기위해서 구동체가 수행해야 하는 동작은 직진과 회전으로 구분할 수 있다. 우선 세 바퀴 구동체가 전/후진 동작을 수행하도록 코딩했는데, 시험 환경에서 모터의 Duty Cycle과 진행 시간에 대응하는 구동체의 속도를 측정하여 마찰력이라는 변수를 상쇄할 수 있었다. 다음으로 장애물을 회피하기 위해 사용되는 회전 알고리즘으로 포인트 턴을 사용했다. 포인트 턴은 두 바퀴가 다른 방향, 같은 속도로 회전할 때 구동체가 회전하는 동작이다. 알고리즘을 테스트 할 때 실제 시험 환경에서 구동체가 라인을 돌면서 자연스럽게 소모되는 배터리 때문에 의도한 회전 각이 나오지 않는 문제점이 발생했다. 우리 팀은 이 문제점을 해결될 때 까지 의도적으로 값을 수정하는 방식으로 해결했다.

## 장애물 인식 및 회피

구동체에 설치되어있는 초음파 센서는 일정한 간격으로 초음파 펄스를 발사하여 돌아오는 시간을 측정한다. 소리의 속도는 섭씨 15도의 환경에서 약 340m/s이고 소리가 장애물에 도달하는 시간을 t초라고 가정한다면 구동체와 장애물 사이의 거리는 (1/2) \* \* 340m/s 로 계산할 수 있다. 이 과정을 실시간으로 반복하여 구동체와 장애물 사이의 거리를 구하게 된다.

구동체가 장애물을 회피하는 동작을 수행하기 전에 얼마만큼 떨어진 거리에서 이 동작을 수행할 지에 관한 기준이 필요하다. 장애물을 회피하는 동작은 구동체와 장애물 사이의 거리가 10cm이하로 가까워 졌을 때 수행하게 된다. 이 기준을 만족했을 때 구동체는 오른쪽 방향으로 포인트 턴을 하여 장애물의 옆 방향을 보게 된다. 이후 장애물이 구동체의 옆에 위치하게 될 때 까지 직진을 수행한 후 왼쪽 방향으로 포인트 턴을 약하게 수행하여 구동체가 라인과 평행하게 위치하도록 조정해준다. 이후 구동체의 뒷바퀴까지 장애물을 지나칠 정도로 직진을 수행한 후 라인이 존재하는 방향인 왼쪽으로 포인트 턴을 수행하여 5방향 트랙킹 센서가 검정 라인을 감지할 때까지 구동체는 전진을 수행하게 된다.

## 곡률 코스 처리

5방향 트랙킹 센서가 감지하는 방향에 대해서 본 보고서는 [00000]꼴의 기호를 사용하게 되고 왼쪽부터 [left most, left less, center, right less, right most]의 방향을 의미한다. 0은 5방향 트랙킹 센서가 검정색을 감지한 즉, 검정선 위에 있는 상황을 의미하며 1은 흰색 즉 검정선이 아닌 곳에 위치한 상황을 의미한다.

구동체가 곡선 구간을 만나면서 수행해야 하는 동작은 5방향 트랙킹 센서의 왼쪽 부분이 감지되어 구동체가 왼쪽 방향으로 주행해야하는 상황 혹은 오른쪽 방향이 감지되어 구동체가 오른쪽으로 주행해야 하는 상황으로 구분 할 수 있다. 이에 대하여 우리가 찾은 경우를 아래의 표[3.3-1]에 정리해 보았다.

표[3.3-1]

|  |  |  |  |
| --- | --- | --- | --- |
| Go Forward : [11011] or [10001] | | | |
| Left Turn | | Right Turn | |
| Strong | Week | Strong | Week |
| [01111] | [10011] | [11110] | [11001] |
|  | [10111] | [11000] | [11101] |
|  | [00111] |  | [11100] |

구동체는 주행 도중 방향을 조정하기 위해서 두 바퀴가 같은 방향으로 회전하지만 회전 속도를 서로 다르게 주어 회전하는 방식을 사용하게 된다. 직진 구간에서 라인을 벗어나면 약간의 방향 조정을 위해 두 바퀴의 Duty Cycle 값의 차이를 작게 설정하게 된다. 하지만 곡선 구간에서는 굴곡이 크기 때문에 두 바퀴의 Duty Cycle 값의 차이를 크게 설정하여 구동체의 방향이 많이 회전할 수 있도록 설정해주게 된다.

# 향후 추진계획

## 향후 계획의 세부 내용

1. S/W적인 측면

소프트웨어를 코딩함에 있어서 기존의 소스는 중복되는 요소가 많았다. 이를 통해 다음과 같은 문제 상황을 마주치게 되었다.

* 코드의 가독성 저하
* 구동체의 의도되지 않은 동작이 빈도 증가
* 팀원들 간의 코드 분석 시간 증가

이 문제를 해결하기 위하여 코드의 설계방식을 지금과 같은 절차 지향 방식의 코드가 아닌 객체 지향 방식으로 재설계 하여 코드의 재 사용성을 증대시키는 방법을 찾을 수 있었다.

1. H/W적인 측면

구동체를 수차례 테스트 해보면서 일부 하드웨어적인 문제점을 찾을 수 있었다. 문제점은 다음과 같다.

* 구동체가 튀어나온 바닥을 지나는 상황에서 간헐적으로 5방향 트랙킹 센서와 연결된 케이블이 분리되는 문제
* 볼트와 너트가 느슨해지는 문제
* 바퀴의 고무 부분에 먼지가 묻은 상황에서 구동체 운전시 미끄러짐 현상 발생

이 문제를 해결하기 위해서 5방향 트랙킹센서의 연결 부분을 절연테이브로 보강하는 작업을 수행 하여 첫번째 문제를 해결하기로 하였고 실험 후 구동체의 볼트를 조여주고 바퀴를 청소해주는 방식의 구동체 유지보수를 하여 다음 번 시험에 지장이 없도록 팀원과 회의를 통해 해결할 수 있었다.

# 애로 및 건의사항

프로젝트를 수행함에 있어서 구동체가 라인 트레이싱을 수행하는 알고리즘을 설계하는 느낌이 아닌 라인을 잘 따라갈 수 있는 적합한 값을 찾아야 한다는 성격이 강했다고 생각한다. 이 값은 절대적으로 구동체를 구동시키는 방식이 아닌 배터리의 잔량에 따른 상대적인 값이므로 연습 환경과 시험환경에서 구동체가 의도하지 않은 동작을 수행하는 경우가 자주 발생했다. 노력의 결과가 시험을 치는 순간에 있어서 결국 무작위로 발생하게 되므로 평가에 있어서 아쉬움이 많이 있었다.