

제20회 임베디드SW경진대회 개발완료보고서

[자유공모]

□ 개발 요약

팀 명	SPACEPEN
	
작품명	스페이스펜(SpacePen)
작품설명 (요약)	어떤 평면이나 공간에서 글이나 그림을 작성한 것을 실시간으로 디지털 이미지로 변환하여 시각화한다. 청각장애인의 소통수단 뿐 아니라 다용도로 사용 가능한 펜이다.
소스코드	https://github.com/siorTeam/smart-pen
시연동영상	https://youtu.be/AMKU_AyEsPM

□ 개발 개요

○ 개발 작품 개요

- 우리의 삶에서 글쓰기는 상당히 큰 부분을 차지하고 있다. 말을 제외한 모든 의사소통은 글로 이루어졌다고 생각하면 훨씬 더 크게 와닿을 것이다. 현대와 같은 스마트폰시대에서는 말보다 글로써 정보를 더 많이 획득하고 공유하고 있다. 통화보다는 메신저를 사용하여 의사소통을 하고 글을 씀으로써 생각을 알리는 일은 우리의 일상이다. 그러나 이러한 일상을 누구나 누릴 수 있는 것은 아니다. 우리가 당연하게 여기는 ‘글쓰기’를 모든 사람들로 하여금 당연해질 수 있도록 고안한 것이 바로 ‘스페이스펜’이다.
- ‘스페이스펜’은 회의 중 청각장애인들을 위한 실시간 의사소통 도구가 있었으면 좋겠다는 생각으로부터 출발하였다. 비장애인 중에는 수화를 할 줄 모르는 사람이 상당수이다. 그러므로 청각장애인 입장에서는 수화를 쓰지 않고 의견을 빠르게 전달할 수 있는 방법이 있다면 유용할 것이다. 그러한 방법 중 하나로 글쓰기를 들 수 있다. 그러나 종이 등의 평면에 글을 쓰는 것은 정면에서 똑바로 글을 쓰고 읽을 수 있고, 쓰고 있는 사람과 가까운 사람만이 볼 수 있다는 공간적 문제가 있다. 따라서 평면뿐 아니라 허공까지 필기의 영역 자체를 넓히고 모든 사람들이 볼 수 있도록 디지털화하여 전송이 가능한 장치의 아이디어가 떠올랐고, 이는 개발 과정을 거치며 ‘스페이스펜’의 핵심 기능으로 발전하게 되었다.
- 본 팀이 고안한 ‘스페이스펜’은 제한된 필기 영역이 없다. 즉 모든 공간 자체가 필기 공간이 될 수가 있다는 점이 이 프로젝트의 핵심이다. 시중에도 비슷한 기능으로 출시된 펜들이 더러 있지만 그 펜이 지원하는 노트나 태블릿 위에서만 필기를 인식할 수 있도록 설계되어 있다.. 저희가 가장 우선순위로 두었던 요소는 펜만으로 필기가 가능하도록 하는 것과 추가적인 필기 영역이 없이 모든 평면에서 필기가 가능하도록 하는 것이었다. 따라서 기존의 태블릿 등에서 사용하는 전자기 유도 원리가 아닌 다른 원리의 필기를 인식하는 방식이 필요하다고 생각했다. 여러 센서들 중 IMU는 앞서 언급한 여러 요소의 구현을 충족할 수 있었고, 이를 이용하여 개발을 진행하였다.
- 또한 본 팀의 ‘스페이스펜’은 노트와 같은 영역에 글씨를 쓰는 것이 아니기에 필기 내용을 실시간으로 확인할 수 있는 방법이 필요했다. 따라서 실시간 처리가 가능하도록 설계 목표를 설정하고 추가적으로 앱을 개발하여 작성한 필기를 실시간으로 확인할 수 있도록 계획을 세웠다.
- 본 팀이 개발한 ‘스마트펜’이 하는 일은 간단하다. 필기를 하고 이를 확인할 수 있게 하는 것이다. 하지만 본 팀은 이 일련의 과정이 어느 누가 사용하더라도 걸림돌이 없으며 누구에게나 편리한 사용자 환경을 제공하기를 원했다. 그래서 그 속에 여러 아이디어를 넣어 이러한 목표를 실현시켰고, 결국 ‘스페이스펜’을 제작 할 수 있었다.

○ 개발 목표

- 본 팀의 ‘스페이스펜’은 어떤 공간상에서 글이나 그림을 작성하면 이를 디지털 이미지로 변환하여 실시간으로 확인할 수 있도록 하는 것이 최종 목표이다.
- 어느 공간에서든지 경로 인식을 할 수 있도록 하는 것이 가장 중요하므로 IMU센서를 활용하는 것이 가장 적합한 선택이었다. 여러 IMU센서(MPU-9250, BNO055, Arduino Nano BLE 33 등)을 실험해본 결과 9축 IMU가 내장되어있는 Arduino Nano 33 ble가 가장 적합함을 확인하였다.

- 또한 실시간으로 필기하는 것이 중요하므로 블루투스 통신을 통해 실시간으로 메인 프로세서에서 작성한 내용을 확인할 수 있도록 계획하였다. 초기 버전의 경우 MATLAB 앱을 통해서, 최종 버전의 경우 Python 앱에서 작성한 필기를 확인할 수 있다.

○ 개발 작품의 필요성

- 본 팀이 개발한 ‘스페이스펜’과 기능이 비슷한 제품은 시중에서도 다수 출시되어 있다. 예를 들어 ‘와콤’과 같이 태블릿과 결합하여 쓰는 펜이나 ‘네오 스마트펜’과 같이 제품과 한 쌍인 특정 노트 및 패널과만 결합하여 쓸 수 있는 펜 등이 있다. 이렇듯 펜과 한 쌍으로 구성된 장치를 통해서만 필기를 할 수 있다는 점은 제약으로 작용하고 있다고 판단하였다. 따라서 본 팀은 특정 평면(태블릿, 노트 등)에 국한되지 않고 필기를 할 수 있다면 방해요소 없이 더 자유로운 필기를 할 수 있을 것이라는 결론을 내렸다.
- 청각장애인이 비장애인과 소통하는 방법은 수화나 글로 할 수밖에 없다. 앞서 언급한대로 비장애인 대부분은 수화를 모르고 글을 쓰면 글의 방향이 쓴 사람의 방향으로 되어 있으며 가까운 사람만 글을 확인할 수 있다. ‘스페이스펜’을 활용하면 필기를 디지털화하여 쓴 사람 뿐 아니라 다른 사람들도 볼 수 있으며 공간에 구애받지 않고 소통할 수 있다.

○ 개발 작품의 활용성 및 기대효과

- ‘스페이스펜’은 본질적으로 청각장애인을 위한 편리한 필기도구를 만드는 것으로부터 시작했고 처음 계획했던 공간 제약이 없는 필기도구의 개발의 목표를 만족하는 작품이다. 따라서 청각장애인들이 좀 더 쉽고 편하게 글을 통해 소통하는 데에 충분히 잘 활용할 수 있을 것이라고 기대한다.
- 또한 개발 후 ‘스페이스펜’을 사용해보니, 다용도로 활용 가능한 도구라는 결론을 내릴 수 있었다. 별도의 필기 인식 장치가 전혀 없이 펜만을 가지고 필기를 할 수 있기 때문에 어느 곳에서나 ‘스페이스펜’만 들고 있다면 기록하고 싶은 것을 바로 필기할 수 있다. 접하는 정보의 양이 많아지고 빠르게 기록하는 게 필수가 되어버린 지금의 시대에 무한한 발전 가능성이 있는 작품이라고 생각된다.

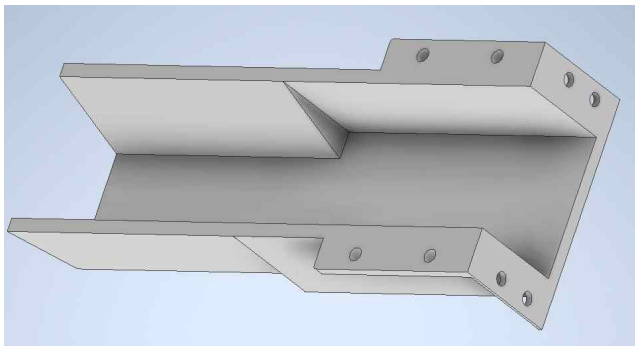
□ 개발 환경 설명

○ Hardware 구성

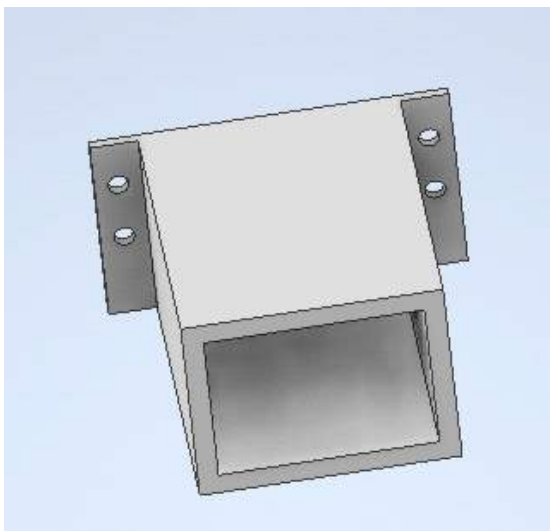
- 팬의 electrical hardware는 관성측정장치, 푸시 스위치, 온보드 프로세서, 통신모듈로 구성되어있다. 관성측정장치는 팬의 가속도 및 지자기 데이터를 측정한다. 푸시 스위치는 팬이 현재 확을 긋고 있는 중인지 판별하기 위함이다. 온보드 프로세서는 관성측정장치와 푸시 스위치에서 보낸 데이터를 받아 메인프로세서로 송신 가능한 정보로 변환한다. 통신모듈은 메인프로세서로 전처리된 가속도 및 지자기 데이터를 전송한다. 관성측정장치, 온보드 프로세서, 통신모듈은 초기 버전에서는 별도의 모듈들로 구성되어있었으나 후에 모든 모듈이 통합되어있는 개발보드로 교체하였다.

- 팬의 mechanical hardware는 크게 케이스와 버튼으로 구성되어있다. 케이스는 개발보드와 전원 공급용 건전지가 내부에 있으며 팬의 모양을 하고 있다. 버튼은 사용자가 팬 내부에 있는 푸시 스위치를 누를 수 있도록 케이스의 외부로 나와 있으며 쉽게 누를 수 있도록 손으로 쥐는 부분과 같은 높이에 있다.

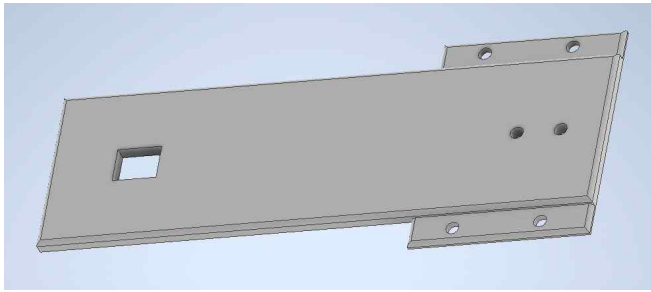
좀 더 세부적으로 보면 (1) 아두이노와 스위치를 부착할 부분, (2)배터리를 넣을 부분, (3) (1), (2)의 위를 덮을 지붕, (4) 스위치를 제대로 고정하기 위한 바구니, (5) 버튼으로 구성되어 있다. 다음은 인벤터로 이들을 설계한 것이다.



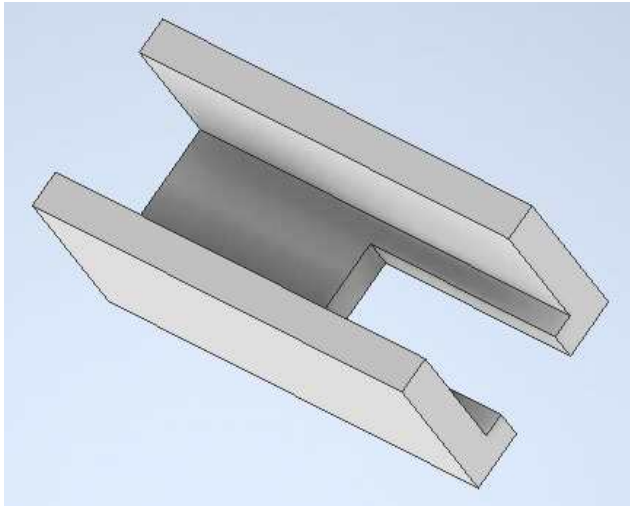
(1) 아두이노와 스위치를 부착할 부분



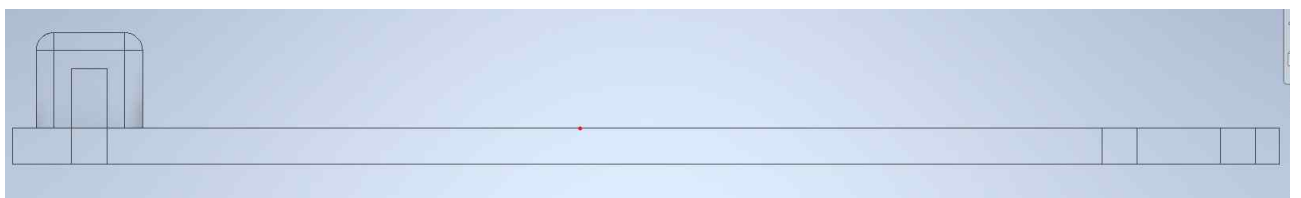
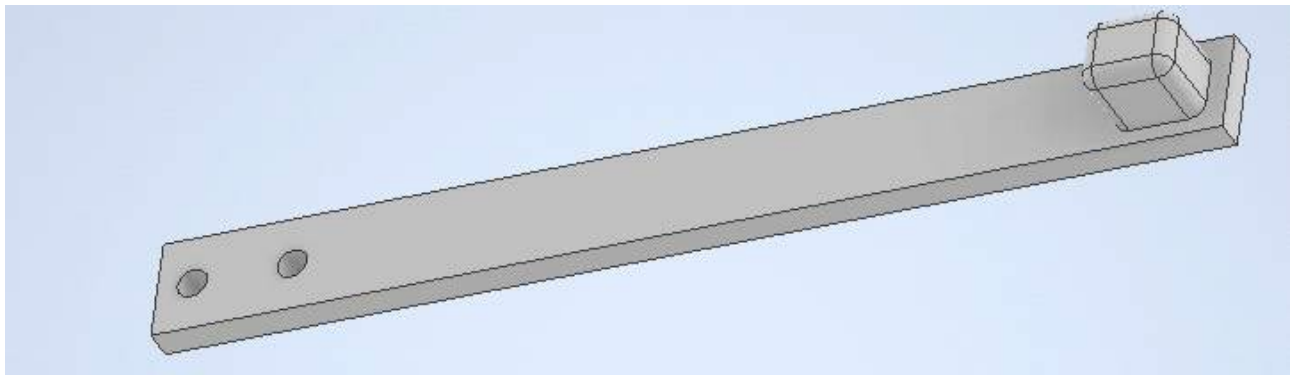
(2)배터리를 넣을 부분



(3) (1), (2)의 위를 덮을 지붕

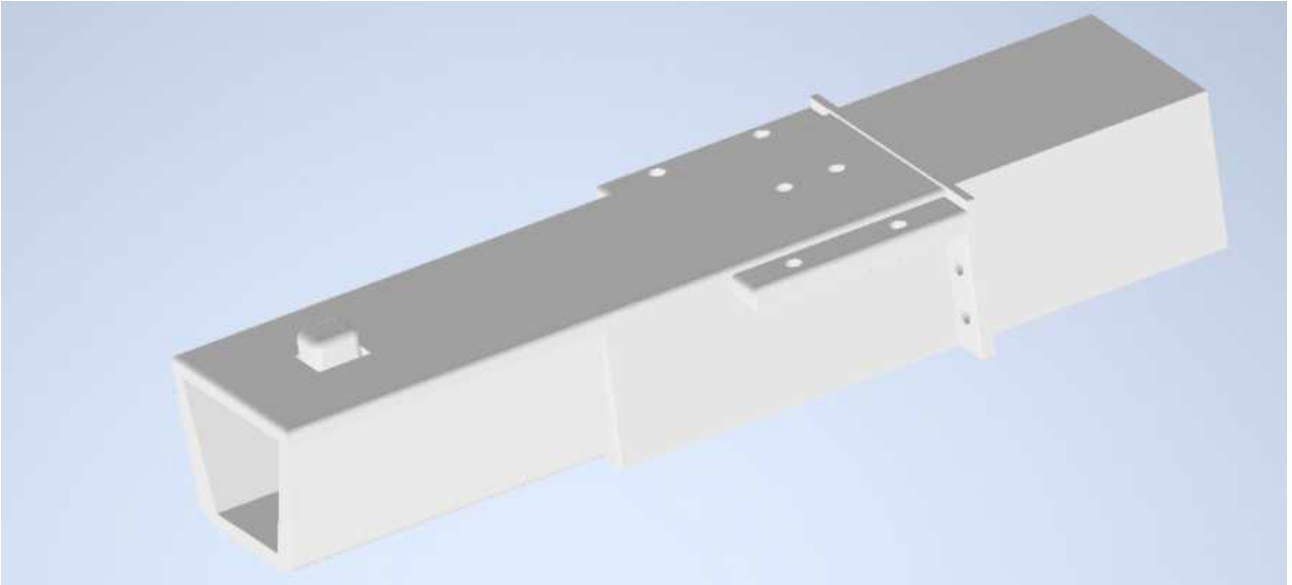


(4) 스위치를 제대로 고정하기 위한 바구니



(5) 버튼

이들을 조립하면 다음과 같은 조립품이 나온다.



○ Hardware 기능 (제어 방법 등 서술)

- 관성측정장치(Inertial Measurement Unit, 줄여서 IMU라고 이하 표기)는 가속도계(Acceleration) 센서와 각속도계(Gyroscope) 센서, 또한 자력계 센서의 조합으로 이루어진 장치로 가속도계와 각속도계의 조합으로 이루어진 IMU를 6축 IMU, 자력계까지 고려하는 IMU를 9축 IMU라고 불린다. IMU는 전체 환경에 대한 자세, 각도 비율, 선형 속도 및 위치를 계산하는 항법 시스템과 통합하여 많이 사용되며 IMU의 센서에서 측정된 데이터를 이용해 메인 프로세서에 보내진 데이터들이 이 항법 시스템의 계산으로 나온 위치를 읽어내는 것을 목표로 하므로 가장 근본적인 데이터를 생산해내는 중요 Hardware이다. 보다 정밀한 측정을 위해 9축 IMU에 해당하는 가속도계, 각속도계, 자력계로 구성된 IMU를 선정하여 진행하였다. 시중에서 구할 수 있는 MPU-6050, MPU-9250, BNO055를 후보로 선정하였고, 차후에 고려한 다른 Hardware와의 구성을 고려하여 최종적으로 LSM9DS1을 최종 IMU로 채택하였다.
- IMU에서 오는 센서의 값들을 해석가능한 수들의 집합으로 구성된 일련의 Bytes데이터로 변환하기 위해 프로세서가 필요하다. 펜에 붙일 정도로 작은 마이크로 프로세서이어야 하고, 테스트하기 적합한 Arduino를 선정하였으며 이는 차후에 시제품 제작에 있어 소형화를 하기 쉽고, 현재 빠르게 테스트를 진행해볼 수 있다는 이점이 있어 Arduino Nano를 선정하였다. 이후 차후에 고려한 Hardware 구성으로 Arduino Nano 33 BLE의 ARM Cortex-M4을 변환 프로세서로 채택하였다.
- 통신모듈은 IMU에서 변환된 데이터들을 메인 프로세서에 해당하는 컴퓨터로 전달해줄 모듈로써 소형에 해당하며 근거리 통신이 지원되어야 하고, 전력이 적게 들수록 좋다. 이 조건에 해당하는 여러 통신들 중에 Bluetooth Low Energy(BLE)와 Wifi가 선정되었는데 BLE의 경우가 전력이 더욱 적게 들고, 근거리 통신에 강점을 가지고 있는 것으로 판단되어 BLE를 통신모듈로 채택하였다.
- IMU와 BLE, 프로세서가 통합되어 있는 개발보드를 시장에서 조사하던 도중, Arduino Nano 33 BLE가 이 주요 Hardware 구성을 통합하고 있음을 확인하였다. 이 보드를 통해 별다른 회로 구성없이 바로 구현할 항법 시스템에 테스트해볼 수 있는 이점과 개발보드임에도 매우 소형화되어 있다는 특징이 있어 이 개발보드에 있는 LSM9DS1와 해당

ARM Cortex-M4 프로세서, BLE를 채택하여 Hardware를 구성하였다.

- 케이스는 CAD 프로그램(Autodesk Inventor Pro)을 사용하여 직접 설계하였다.

[이 부분은 설계 당사자가 써주는 거가 좋을 것 같음]

[일단 내가 알기로 실제 전자적 하드웨어가 들어가는 베이스랑]

[그 베이스를 덮는 덮개랑 뒷부분에 배터리 들어가는 홀더 파트랑]

[펜의 앞부분을 구성하는 원뿔모양이랑 캔틸레버 구조의 버튼이랑]

[이런거들 있잖아 그거를 좀 써주면 될 듯]

[순서는 먼저 구성 요소를 이름만 쪽 나열하고]

[각 구성 요소가 이 펜을 이루기 위해서 어떤 역할을 하는지 간략하게 설명하면]

[좋을거 같음. 일종의 조립설명서를 적는다고 생각하면 편하겠다]

[전자적 하드웨어에 푸시 스위치는 안썼는데 스위치 하나 설명한다고]

[한 문단 쓰는데 겁나 웃긴거 같아서 뻘음. 기계적 하드웨어 버튼 설명할 때]

[대충 언급만 해줘도 될 듯]

○ Software 구성

- Arduino에 이식된 Software

Arduino에 이식된 소프트웨어는 IMU와 BLE에 대한 데이터 처리를 위해 IMU의 센서의 데이터를 읽어오는 부분과 BLE로 송수신하는 부분으로 구성된다. Arduino Nano 33 BLE에 내장된 IMU값을 읽어올 수 있게 Arduino_LSM9DS1 라이브러리를 이용했고, BLE 설정과 송수신 처리를 위해 ArduinoBLE 라이브러리를 이용했다

- MATLAB 앱

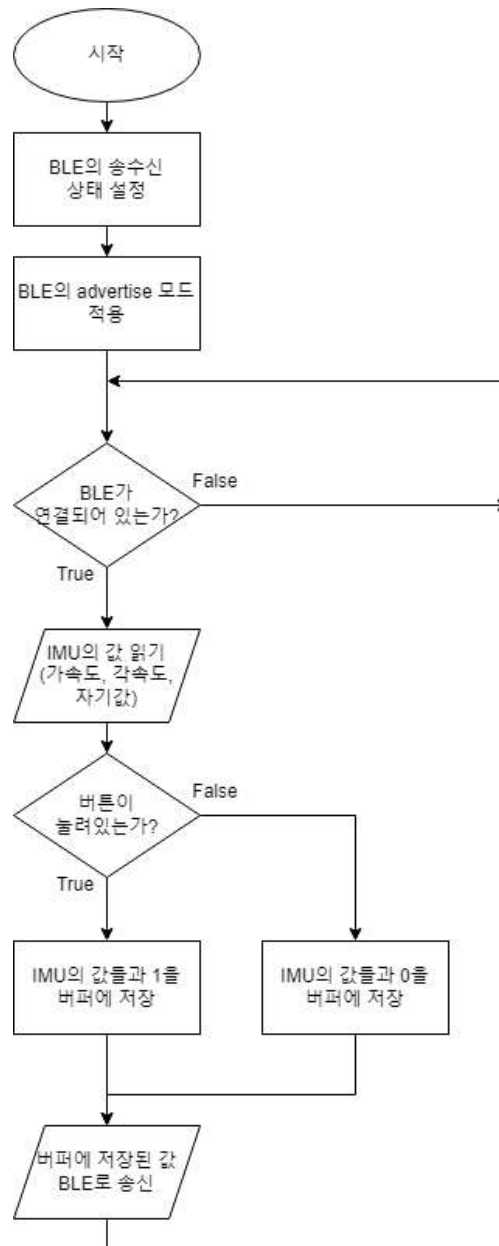
개발초기에 알고리즘을 검증하고 경로추적의 정확도를 높이기 위해서 제작된 프로토타입 앱이다. 시리얼을 이용해서 IMU 센서로 부터 값을 받아오고 필터 라이브러리를 이용해서 경로를 추정할 수 있다. 이때 필터의 파라미터를 쉽게 조절해가며 출력값을 비교하기 쉽기 때문에 해당 앱을 이용하여 적절한 파라미터를 조절할수 있다. 앱 내부에서 사용한 Madwick filter, Mahony filter와 같은 합성 알고리즘은 해당 필터를 제안한 Sebastian Madgwick이 직접 제공한 MATLAB 라이브러리를 이용했다. 그리고 이를 이용해서 경로를 재구성하는 코드는 X-IO tech에서 제공한 오픈소스중 Oscillatory motion tracking with x-IMU의 코드를 활용했다.

- Python 앱

프로토타입앱에서 추정한 필터를 통해 최종적으로 최적화한 앱이다. 시리얼을 이용해서 IMU 센서로부터 오는 데이터를 받아온다. 이 값들을 필터 라이브러리를 이용해 최종적으로 경로를 그래프를 그리고 이를 이미지로 저장할 수 있다. 이 앱에서 사용한 Madwick filter는 해당 필터를 제안한 Sebastian Madgwick이 직접 제공한 Python 라이브러리를 이용했다. 이를 이용한 경로 구성은 AHRS 라이브러리를 이용했다.

○ Software 설계도 (흐름도 및 클래스 다이어그램 등 / 개발언어에 따라 선택)

– Arduino에 이식된 Software



아두이노에 이식된 소프트웨어 전체 흐름도

– MATLAB 앱

1) /@MadgwickAHRS, /@MahonyAHRS

각각 Madwick 필터, Mahony 필터 클래스를 정의 한 스크립트 파일이 있다. MATLAB 코드에서는 각각의 클래스를 객체화 하고 Update 메소드를 이용해서 본인의 orientation을 추정한다.

2) quaternion_library

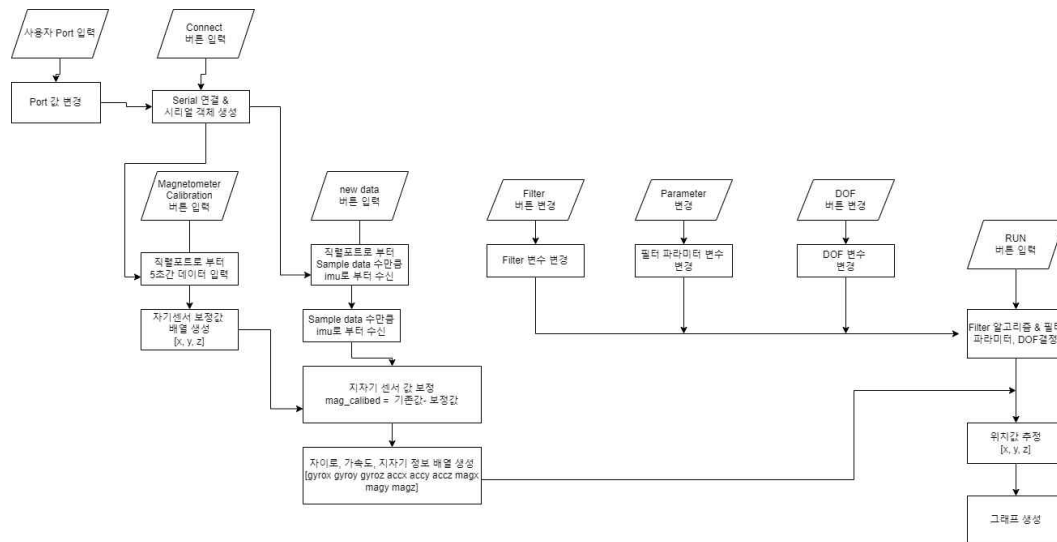
Madwick, Mahony필터에서는 본인의 orientation을 계산하기 위해서 내부에서 방향을 quaternion으로 표현하고 계산한다. 해당 폴더 내부에는 quaternion의 곱셈등의 연산과 quaternion을 Rotation matrix로 변환해주는 코드가 포함되어있다.

3)GetSampleFrequency.m

위치를 추정하기 위해서 데이터가 몇초간격으로 샘플링되는지 알아야 한다. 해당 스크립트에서는 300개의 데이터를 받는 시간을 구하고 받은 데이터의 개수에서 소요된 시간을 나누어서 샘플링 속도를 구하였다.

4) evalkit_app_forBLE.mlapp

MATLAB 앱디자이너로 구현한 프로토타입 앱이다. 해당 앱을 이용하여 파라미터를 조절해가면서 경로 추정 과정을 확인 할 수있다.



matlab앱 전체 흐름도

- Python 앱

1) madgwickahrs.py

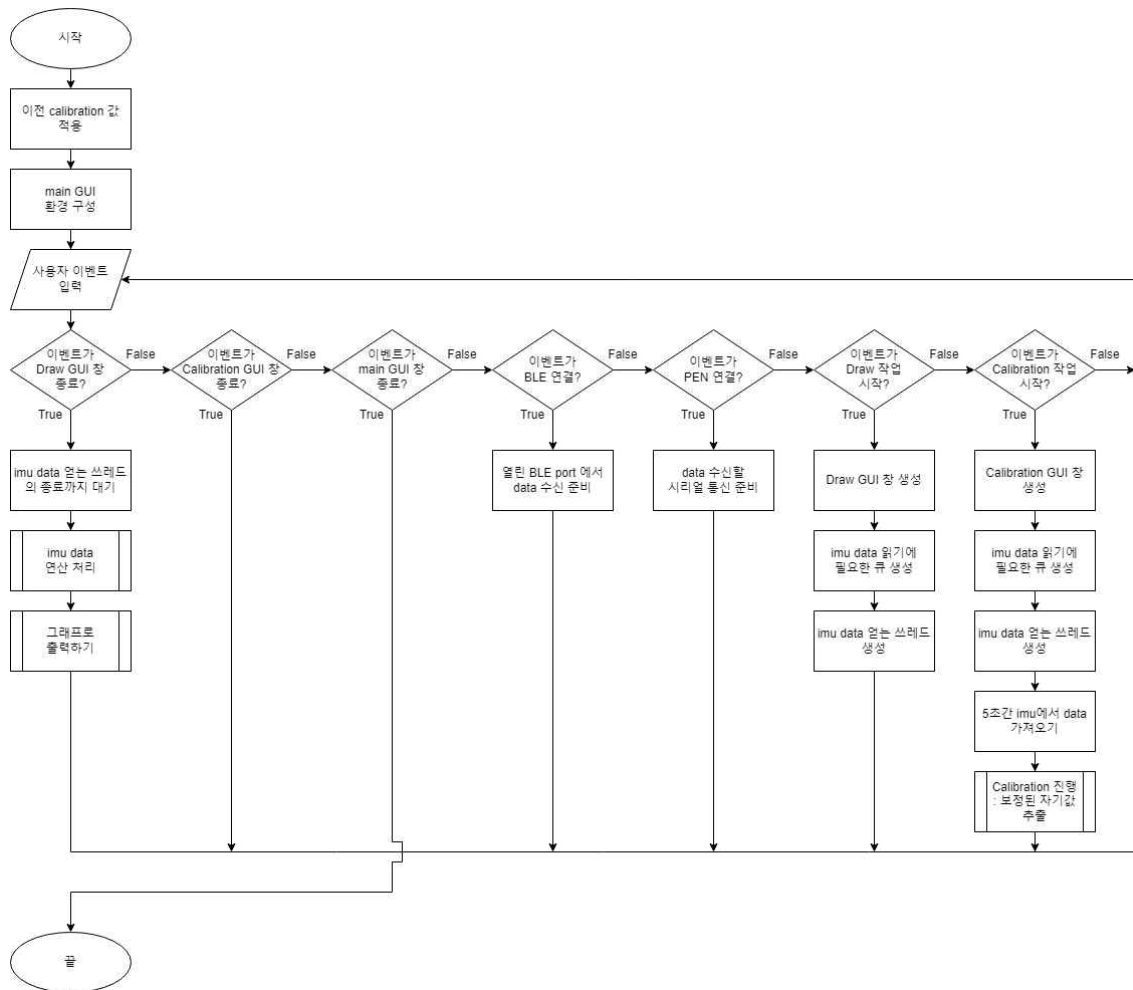
Madwick 필터를 정의한 라이브러리이다. 해당 필터를 객체로써 사용할 수 있고 입력되는 센서 데이터들을 통해 orientation을 추정할 수 있다.

2) quatern2rotMat.py

Madwick필터 내에서는 필터의 orientation을 계산하기 위해서 좌표의 방향을 quaternion으로 표현하고 계산한다. 이 라이브러리는 위치 추정을 위해 가속도값을 좌표변환하는 변환 행렬을 만들어주는 코드가 포함되어 있다.

3) main.py

필터 연산과 좌표변환을 위한 라이브러리를 제외한 GUI로 제어하기 위한 소스코드이다. 해당 코드를 실행하여 동작한 GUI를 이용하여 SPACEPEN의 경로를 확인하고 이를 이미지로 저장할 수 있다.



Python 앱의 전체 흐름도

○ Software 기능 (필요 시 알고리즘 설명 포함)

– Arduino

해당 소프트웨어에 이식된 소프트웨어에는 다음과 같은 동작을 수행한다.

1) IMU 센서값 읽기

내장된 IMU의 가속도계, 각속도계, 자기계 센서 각각의 3개의 값, 총 9개의 값을 읽어와 데이터로 구성한다.

2) BLE를 통해 송수신

내장된 BLE를 통해 Bluetooth 통신을 가능하게 해준다. 데이터를 수신할 메인 프로세서의 연결과 연결을 위한 설정, 일련의 문자집합 형태로 변환된 센서들의 값들과 버튼의 활성화 여부가 이 통신을 통해 전달된다.

– MATLAB

해당 앱을 이용해 다음과 같은 동작을 수행 할 수있다.

1) 시리얼 연결

MATLAB의 serial 객체를 생성하여 아두이노와 시리얼 연결을 구성한다.

2) 데이터 입력

시리얼 포트로부터 입력되는 센서 데이터를 읽는다. 읽는 데이터의 개수는 Sample Data의 숫자를 입력하여 변경가능하다. 읽은 데이터는 곧바로 MATLAB에서 사용할수 있는 데이터 배열의 형태로 변형된다.

3) 지자기 센서 보정

자기 센서의 Hard iron distortion을 보정할 수 있다. 보정값은 데이터 배열의 형태로 저장되며 보정값은 데이터를 입력 받아 데이터 배열을 생성하는데에 사용된다. 그리고 보정값 데이터는 앱이 실행될 때마다 초기화된다.

4) 필터 파라미터 변경

데이터를 입력 받은 후 RUN버튼을 눌러서 경로추정 알고리즘을 수행하기 전에 그 과정에서 사용되는 여러가지 변수를 변경할 수있다.

DOF: 사용할 센서의 종류

Filter: 사용할 필터의 종류 (Madwick filter or Mahony filter)

Ki, Kp : Mahony 필터에서 사용하는 파라미터

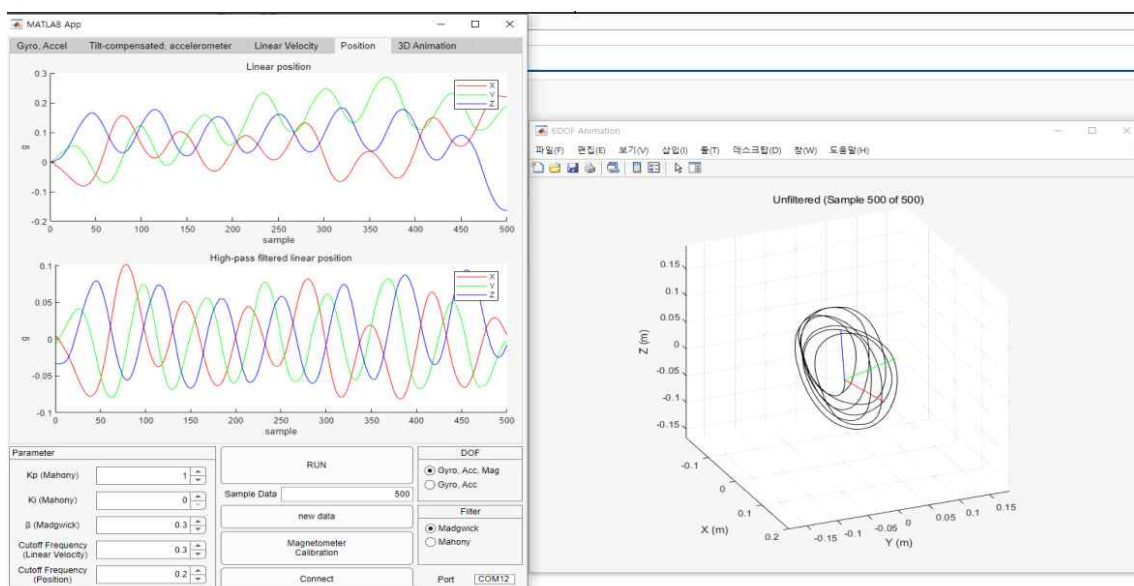
β : Madwick 필터에서 사용하는 파라미터

cutoff frequency : 경로값 계산과정에서 생성되는 drift를 제거 하기 위해서 사용하는 High-pass filter의 차단 주파수

5) 경로 추정 및 과정 시각화

최종적으로 추정된 경로를 애니메이션으로 그린다. 그리고 필터를 이용해서 본인의 방향을 추정하고 중력가속도를 제외한 가속도의 계산, 그리고 그 가속도를 적분해가면서 위치를 추정하는 과정 중간중간의 값들을 그래프로 그려서 위치 추정에서 오류가 난 부분을 파악 할 수있다. 확인 할 수 있는 그래프는 다음과 같다.

센서로 부터 받은 값(자이로, 가속도계, 자자기계), 중력 가속도를 제외한 가속도 값, 속도와 High-pass filter를 거친 속도값, 위치값과 High-pass filter를 거친 위치값 앱을 실행하여 위치를 추정하면 다음과 같은 결과창을 볼 수있다.



- Python

1) 블루투스 연결

메인 프로세서에 해당하는 컴퓨터의 블루투스 연결을 준비하는 기능이다. Arduino의 블루투스 와 연결하기 위한 설정과 구성을 진행한다. 송신한 데이터들을 변환할 수 있게 채널을 구성하고 동기화하는 작업이 이루어진다.

2) 시리얼 연결

위를 통해 연결된 블루투스 채널에서 수신한 데이터들을 python 환경에서 변환하고 처리할 수 있게 구성한다.

3) 지자기 센서값 보정

자기 센서 값들의 Hard iron distortion을 보정할 수 있다. 보정값은 데이터의 배열의 형태로 생성되어 이후 재실행하였을 때 반영할 수 있도록 별도의 파일(mag.txt)로 저장된다. 그리고 보정을 재실행할 때마다 보정값을 초기화한다.

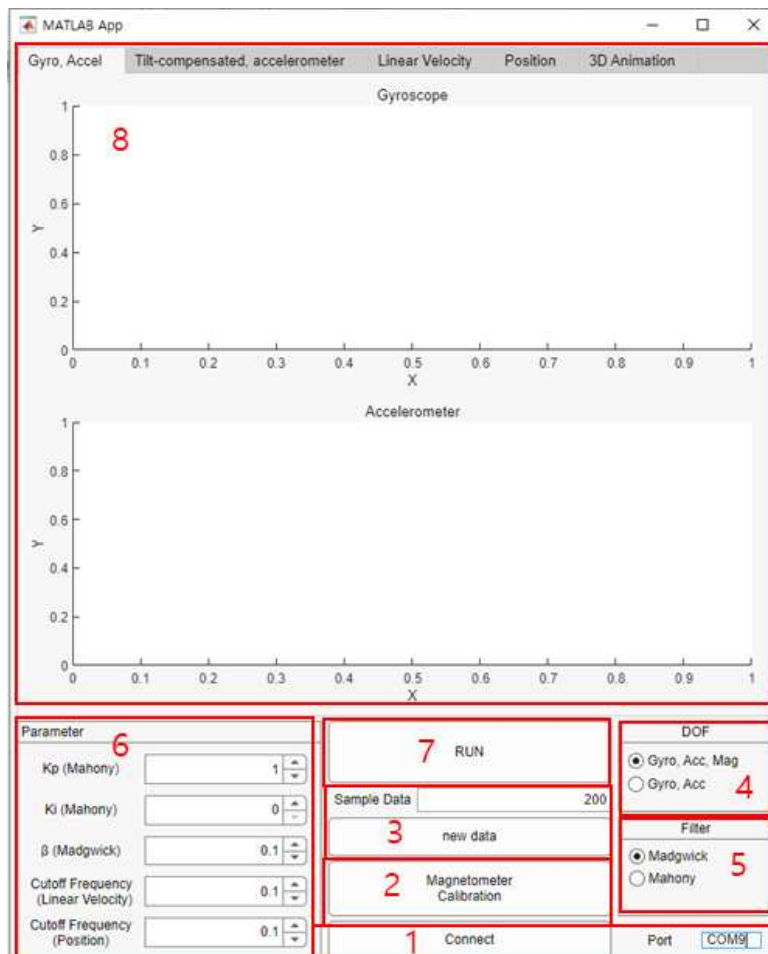
4) 경로 추정 및 시각화

최종적으로 필터를 적용한 좌표변환을 통해 추정된 경로를 그래프로 그린다.

○ 프로그램 사용법 (Interface)

- MATLAB 앱

smart-pen/week6/Matlab code/evalkit_app_forBLE.mlapp를 MATLAB을 이용하여 실행한다.



1. 시리얼 연결

앱을 열었다면 가장 먼저 아두이노와 BLE로 연결을 해야 한다. 먼저 Port칸에 시리얼 포트번호를 입력후 'Connect'를 누른다. 이번 작품에서 사용한 ble-serial을 통해 ble와 pc사이에 직렬포트를 생성했다면 직렬포트는 COM9 가 될것이다.

2. 자기센서 보정

지자기 센서의 캘리브레이션을 하는 기능이다. Magnetometer Calibration 버튼을 누르고 imu센서를 모든 축 방향으로 회전시킨다. 앱 실행후 최초 한번 실행하여 보정값을 얻어준다.

3. 데이터 얻기

Sample Data 입력 칸에 imu로 부터 얻기 원하는 값의 수를 정하여 입력한다. 그리고 new data버튼을 누르면 imu로 부터 데이터를 얻어온다. 얻은 데이터는 다시 new data를 클릭하거나 앱을 재실행하지 않는다면 유지된다.

4. DOF

경로추정시 사용할 센서값의 개수를 정한다. Gyro, Acc, Mag를 선택할 시 센서합성에서 자이로, 가속도계, 지자기 센서를 사용하며 Gyro, Acc를 선택할 시 자이로, 가속도계만을 이용한다.

5. Filter

센서 합성에서 사용할 필터를 선택한다. 사용가능한 필터는 Madwick 필터 와 Mahony 필터가 있다.

6. Parameter

센서 합성에서 알고리즘에서 사용할 파라미터값들을 결정한다.

1)Kp, Ki : Mahony 필터 사용시 조절

2) β : Madwick 필터 사용시 조절

3) cutoff frequency : 경로를 구성할때 속도와 위치의 drift를 제거 하기 위해 high-pass 필터가 linear velocity값에 한번, position값에 한번 쓰인다. 각 각에서 사용되는 cutoff frequency를 조절한다.

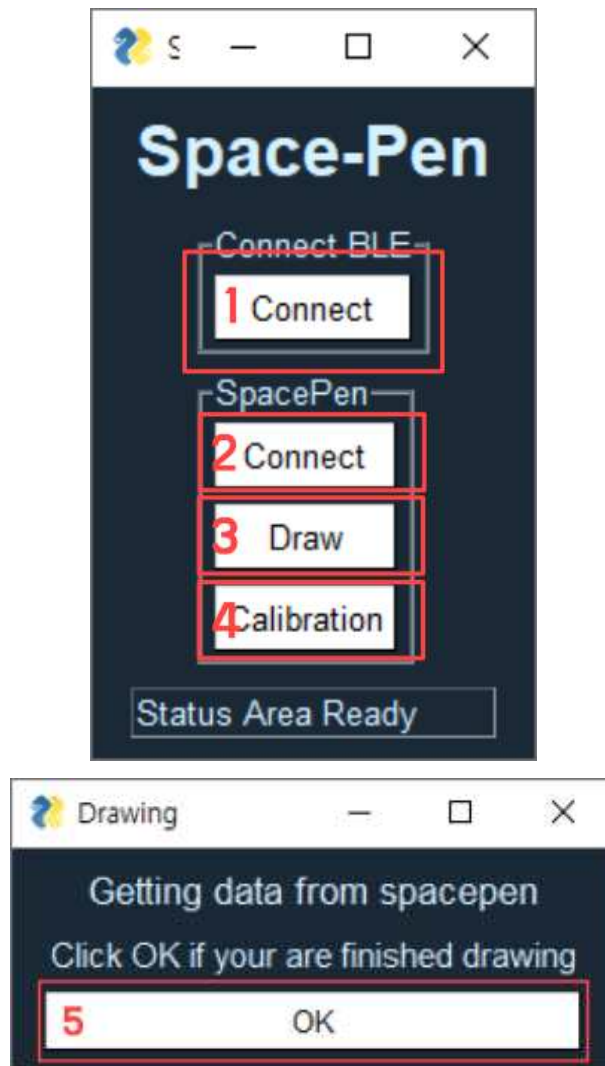
7. RUN

new data버튼을 눌러 데이터를 읽었고 필터와 파라미터 설정까지 완료 되었다면 RUN 버튼을 눌러 경로 추정을 시행한다. 시행 결과 추정된 경로가 그려지고 상단의 그래프에서는 경로 추정에서 사용된 값들을 시각화 해준다. 새로운 데이터를 받지 않고 필터 파라미터만 바꾸어 가며 여러번 경로 추정을 시행할 수있다

8. 그래프

자이로, 가속도, 선형가속도, 위치 값등 센서를 합성하고 high-pass filter를 적용하는 과정 사이의 값들을 그래프로 시각화 해준다. 필터의 파라미터를 조절 하는 과정에서 활용 된다.

- Python 앱



1. 블루투스 연결

Connect를 누르면, 새 프롬프트 창이 열리면서 목적으로 하는 Arduino가 주위에 있는지를 직접 스캔하여 발견되었을 경우 양방향 통신이 가능하게 채널을 구성하고 동기화하는 작업이 자동으로 수행한다.

2. 시리얼 연결

해당 Connect를 누르면, 구성된 블루투스 채널을 python에서 처리할 수 있게 시리얼과 연결한다.

3. 데이터 읽기 시작

Draw버튼을 누르면 아두이노의 블루투스로부터 데이터를 읽어오기 시작한다. 해당 데이터들은 저장되어 이후 그래프 그리기에 반영된다.

4. 자기 센서 보정

지자기 센서의 값의 오차를 줄이기 위해 캘리브레이션하는 기능이다. 해당버튼을 누르고 아두이노를 모든 축 방향으로 회전시킨다. 그러면 보정된 자기값이 추출돼 이후 그리는 그래프에 반영된다.

5. 그래프 그리기

OK버튼을 누르면 그동안 입력받았던 데이터들을 필터를 적용한 경로를 추정하여 최종적으로 해당 경로를 그래프로 그려 시각화해준다. 시각화가 완료될 경우, 그래프가 그려진 캔버스 창이 열리며 SPACEPEN으로 그린 이미지가 보여진다. 이 이미지는 마우스 드래그를

통해 보이는 방향을 전환하는 것도 가능하다.

○ 개발환경 (언어, Tool, 사용시스템 등)

– MATLAB 앱

MATLAB 스크립트와 MATLAB 앱디자이너를 이용해서 앱을 제작했다.

Toolbox: Sensor Fusion and Tracking Toolbox

– Python 앱

PySimplrGUI 프레임워크를 이용하여 python 개발환경에서 GUI를 구현하였고,
pyserial 라이브러리를 통해 블루투스에서 오는 데이터를 수신하였다.

□ 개발 프로그램 설명

○ 파일 구성

- 1) madgwickahrs.py : 필터 라이브러리
- 2) quatern2rotMat.py : 좌표 변환 행렬 생성 라이브러리
- 3) main.py : 파이썬 실행하는 메인코드
- 4) nano33ble_v2.ino : 아두이노에 이식한 코드

○ 함수별 기능

전역변수

-PORT_OBJ ; dictionary

-PORT_STOP_THREAD ; boolean expression

<search_port_list>

● 인자: void

● 기능:

1. 주변 device를 검색한다.

2. 검색된 device의 vid, pid, serial_number를 PORT_INFO란 변수에 각각 list 형태로 넣는다.

e.g. PORT_INFO[device1]=[pid, vid, serial_number]

와 같이 2차원 배열 형태로 정보를 저장한다.

3. 특정 device들의 이름만을 append함수를 통해 따로 tar이란 list에 저장한다.

e.g. tar = [device1, device2...]

● return: tar: list

<get_port_info>

● 인자: _port_id: str

● 기능: search_port_list 함수를 통해 찾은 device들 중 특정 device의 vid, pid, serial_number를 얻을 수 있다.

● return: f"VID:{PORT_INFO[_port_id][0]}\nPID:{PORT_INFO[_port_id][1]}\nSN:{PORT_INFO[_port_id][2]}" :str

<start_serial_data>

● 인자: _port_id: str, _baud_rate: int, _stop_thread: bool, _queue: queue.Queue

● 기능: serial 모니터를 열고 serial에 출력되는 데이터들을 readline()을 통해 읽는다.

● return: void

<process_imu>

● 인자: _queue : queue.Queue, mag_calib: list

● 기능: imu 센서로부터 받아온 가속도값, 지자기값, 자이로값들을 매드윅 필터를 통해 보정함.

- **return:** points, btn_arr

<plotter>

- **인자:** points, btn_arr
- **기능:** process_imu 함수를 통해 얻은 points와 btn_arr를 시각적으로 보여준다.
- **return:** void

<make_BLEport>

- **인자:** void
- **기능:** 파이썬에서 cmd를 통해 serial 모니터의 imu 센서값들을 ???????????????
- **return:** void

<start_serial>

- **인자:** port_id: str, _baud_rate: int
- **기능:** serial 모니터에 device의 port_id와 _baud_rate를 띄워준다
- **return:** serial.Serial(_port_id, _baud_rate)

<get_imu_data>

- **인자:** ser, data_queue, port_queue
- **기능:** serial 모니터의 data값들을 읽고 data란 변수에 저장
- **return:** void

<magnetometer_calib>

- **인자:** _queue: queue.Queue
- **기능:** 자기값 보정 후 보정된 값을 magx_correction, magy_correction, magz_correction 각각에 저장
- **return:** [magx_correction, magy_correction, magz_correction]

<make_main_win>

- **인자:** 없음
- **기능:** 사용자가 쓸 전체적인 윈도우 창 생성
- **return:** sg.Window('Smart-pen', LAYOUT, resizable=True, finalize=True, element_justification='center')

<make_draw_win>

- **인자:** 없음
- **기능:** OK 버튼을 누를 시 pen으로 그림을 그리고 데이터를 가져올 수 있도록 윈도우 실행 환경 구성
- **return:** sg.Window('Drawing', layout, resizable=True, finalize=True, element_justification='center')

<make_calib_win>

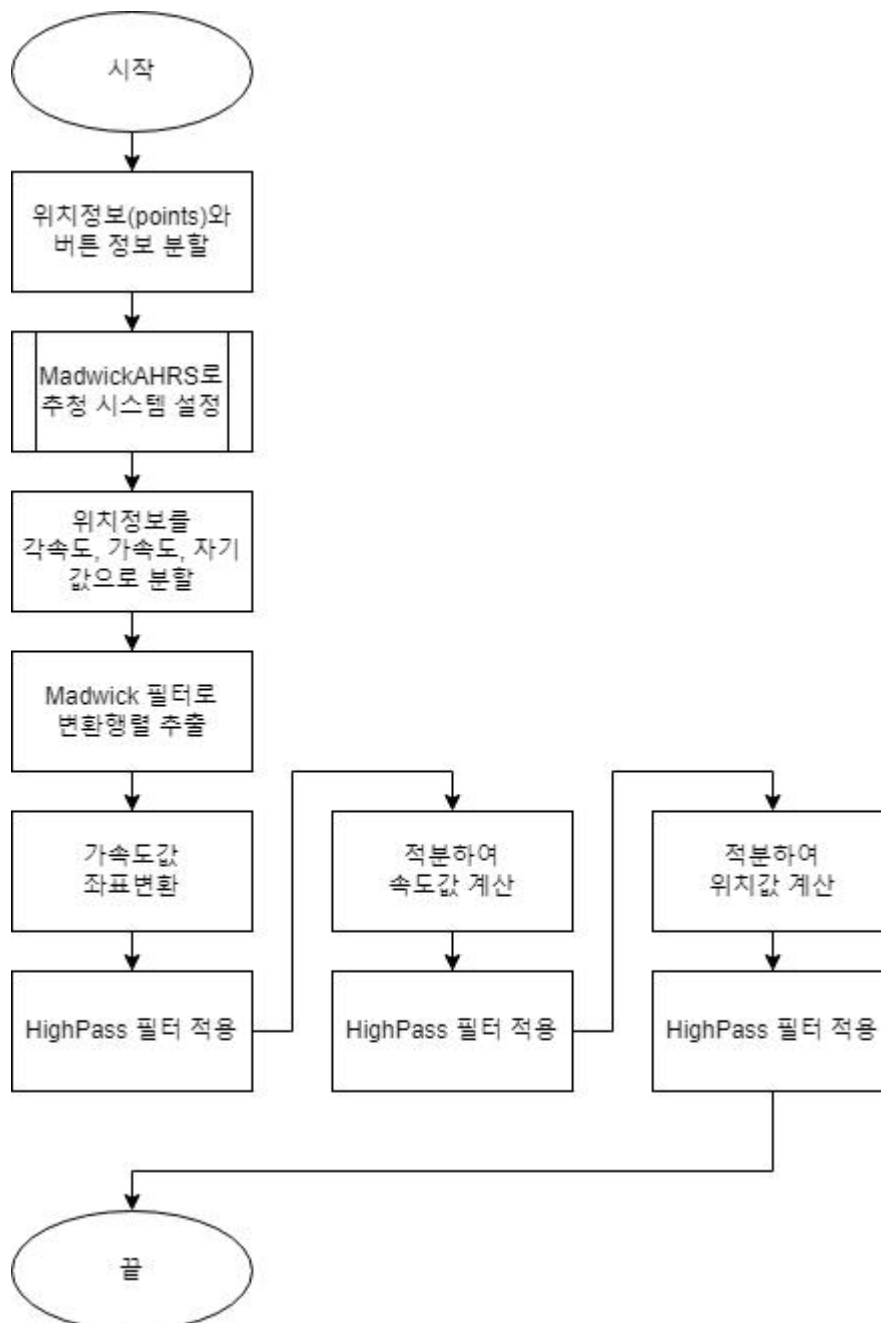
- 인자: 없음
- 기능: 사용자로 하여금 펜을 돌리도록 지시하면서 calibration을 진행할 수 있게 도와주는 윈도우 실행환경 구성
- return: sg.Window('Calibrationg', layout, resizable=True, finalize=True, element_justification='center')

<quaternion_rotation_matrix>

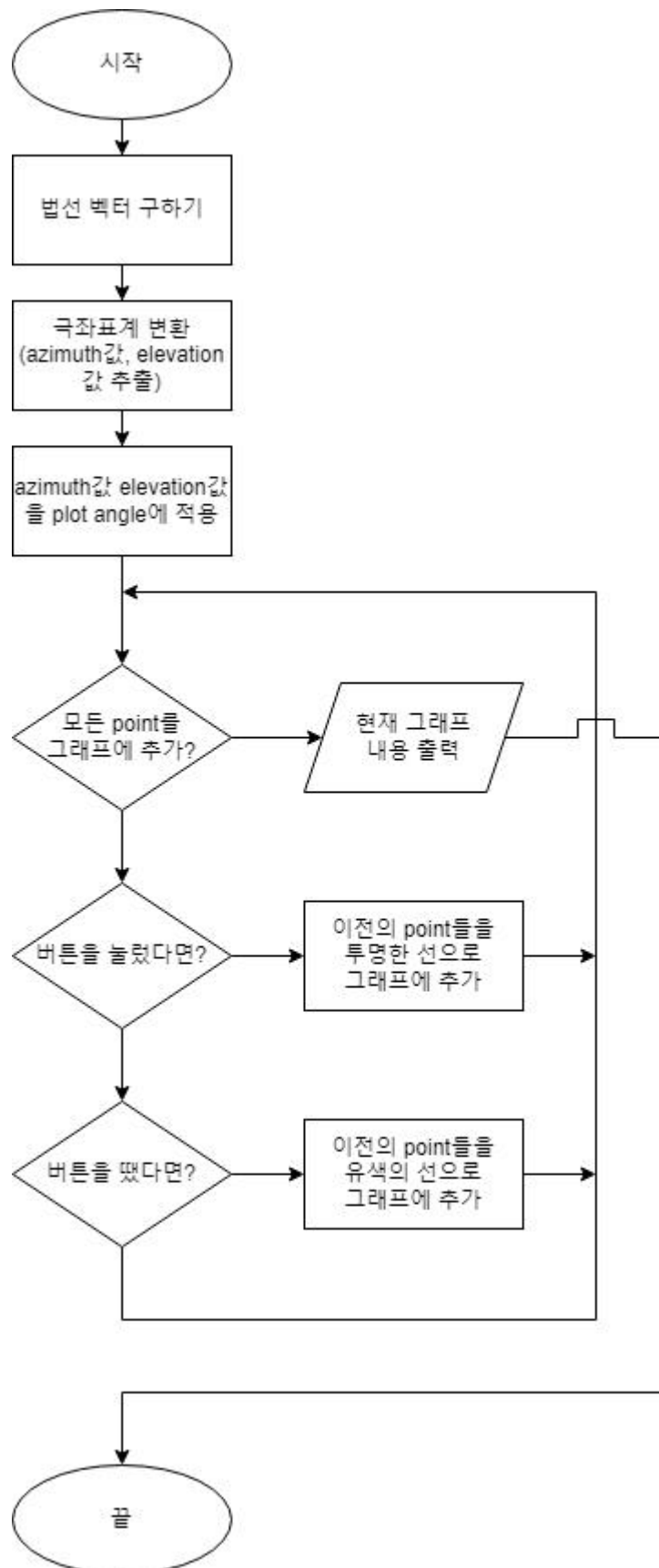
- 인자: Q
- 기능: 4*4 좌표계를 변환하여 쿼터니안 연산을 수행한 후 다시 원래 좌표계로 변환해준다.
- return: rot_matrix

○ 주요 함수의 흐름도

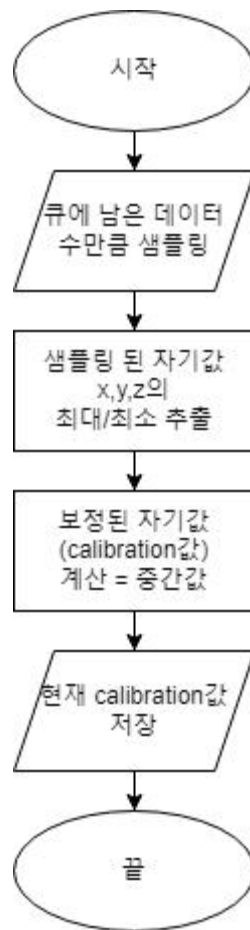
1) imu data 연산처리 : <process_imu>



2) 그래프로 출력하기 : <plotter>



3) Calibration 진행 : <magnetometer_calib>



○ 기술적 차별성

- 평면에 관계없이 3D공간에서의 경로를 직접 추정하여 이를 2D평면상에 투영하여 추버할 수 있는 기술의 차별성이 있다. 이는 샘플링된 데이터의 법선벡터를 구해 현재 데이터들을 투영할 평면을 추정하여 이를 그래프로 그릴 수 있다는 특징이 있다.

또한 추가적으로 방향을 캔버스에 조정할 수 있어 이를 원하는 시점으로 커스텀하는 것이 가능하다는 차별성또한 존재한다.

□ 개발 중 발생한 장애요인과 해결방안

○ 스페이스펜의 위치 추정의 정확성이 떨어지는 문제

- 본 팀의 작품은 가속도 및 지자기 데이터를 필터에서 센서 퓨징하여 펜의 위치를 추정한다. 초기 버전은 부정확한 IMU 데이터, 낮은 샘플레이트, 튜닝되지 않은 필터로 인해 위치 값에 드리프트가 발생하여 경로가 밀려서 출력되는 문제가 있었다. 이는 하드웨어적 접근과 소프트웨어적 접근을 통해 해결하였다.
- 초기 버전에서는 MPU-9250의 낮은 정확도와 ATmega328의 낮은 성능으로 인해 데이터 샘플레이트가 하락하는 문제가 있었다. 이 문제를 해결방안으로 IMU센서 (LSM9DS1)와 ARM Cortex-M4기반 마이크로 컨트롤러, 그리고 BLE통신 모듈이 통합된 Arduino Nano 33 BLE 모듈로 하드웨어를 변경하였다. 이를 통해 IMU 데이터의 노이즈 감소, 16hz에서 64hz로의 샘플레이트 대폭 향상을 이루었다. 또한 하드웨어 구성에 필요한 모듈 3개가 통합된 모듈로 회로 구성 및 개별 모듈간 통신 속도 측면에서도 효율적이었다.
- IMU 데이터로부터 orientation을 구할 때 정확도를 높이기 위해 필터를 사용하게 되는데 초기 버전에서는 Mahony filter를 이용하였다. 하지만 해당 필터로는 본 팀이 의도한 정확한 위치추정이 불가능하였다. 따라서 필터를 Mahony filter에서 Madgwick filter로 변경하여 지자기 데이터를 실시간으로 이용하도록 하였다. 또한 해당 필터의 특성을 결정하는 파라미터 값을 쉽게 조절할 수 있는 MATLAB 앱을 개발하였다. 이 앱으로 파라미터 값을 바꿔가며 센서 데이터가 필터를 거쳐 변환되는 과정을 시각화 할 수 있었고, 이 과정을 반복하여 스페이스펜 모듈에 가장 적절한 파라미터를 구할 수 있었다.

○ BLE 통신속도

- 메인 프로세서(PC)가 스페이스펜과 통신하기 위해서 bleak라고 하는 BLE통신 Python 라이브러리를 이용했다. 그러나 해당 라이브러리 이용 시 IMU값을 읽는 속도가 6hz로 매우 낮았다. 프로토타입 버전의 경우 serial 통신으로 데이터를 송수신하는데 이 때는 PC의 serial 버퍼에 값이 저장되고 스페이스펜에서는 지속적으로 데이터를 전송하기에 60hz의 높은 통신 속도가 가능했지만, BLE 통신의 경우 버퍼의 부재로 인해 생긴 문제로 보인다. 이를 해결하고자 ble-serial 라이브러리를 이용해 PC에서 가상의 serial 포트를 열어 BLE통신이 이루어지도록 하여 전송속도를 다시 60hz로 높일 수 있었다.

○ 3d 경로의 2d 평면으로의 projection

- 펜으로 경로를 그리면 3d 값을 갖는데 이를 2d로 변환해주기 위해서는 경로 위 세 점을 찍어 평면을 만들고 그 평면 위로 projection하는 방식으로 출력하게 된다. 이 과정에서 경로에 좌표변환을 하게 되는데 이를 위한 inverse transform 연산 자체가 자원 소모가 크고 평면의 법선벡터만 알 뿐 3차원 좌표계를 정의하기 위한 나머지 두 벡터는 임의로 생성해줘야 했다. 이 문제는 플로터 라이브러리인 matplotlib의 뷰포인트 변환함수를 이용하여 해결하였다. 법선벡터를 극좌표계로 변환하여 얻은 elevation angle과 azimuth angle을 이용하여 경로를 화면에 띄우는 각도를 조정하여 모니터에 대해 2d projection이 되도록 프로그램을 작성하였다.

□ 개발결과물의 차별성

○ 스타일러스 펜

- 애플의 애플펜슬, 삼성의 S펜 등 태블릿 기기와 같이 사용하는 스타일러스는 본 팀의 작품과 같이 사용자의 필기를 디지털화할 수 있다. 그러나 필기 범위와 무게, 궁극적으로는 이동성 측면에서는 본 팀의 작품이 더욱 유리하다.
- 태블릿의 스타일러스는 필기 범위가 태블릿의 화면 크기에 구속되는 반면 본 팀의 작품의 경우 경계제한이 없기 때문에 이론적으로 무한한 크기의 도형을 그릴 수 있다. 무한한 평면 기능은 시중에 판매되는 대부분의 태블릿 기기도 지원하는 기능이지만 필기 면적이 무한한 것일 뿐 실제 필기의 획이 무한하지는 못하다. 태블릿 화면보다 더 큰 도형을 그려야 하는 경우 작게 도형을 먼저 그리고 별도의 이미지 편집기능을 통해 크기를 늘려야 한다. 본 팀의 작품은 별도의 과정 없이 바로 원하는 크기로 그리면 되므로 훨씬 편리하다.
- 스타일러스는 태블릿 장치 위에만 필기가 가능한데 만약 벽의 방향으로 필기를 해야 한다면 태블릿을 벽을 향해 들고 그 위에 필기해야 한다. 필기하기 위해서는 태블릿 기기와 펜을 모두 지니고 있어야 하는데 태블릿 기기는 보통 크기가 큰 만큼 부피를 많이 차지하며 무게가 무겁기 때문에 이동성이 떨어진다. 본 팀의 작품은 필기 평면 역할을 하는 별도의 장치가 없기 때문에 펜만 벽의 방향으로 들고 필기할 수 있다. 또한 현재 구현되지는 않았지만 스마트폰 앱이나 웹앱 등으로 프로그램을 이식한다면 이론적으로 블루투스 4.0(구체적으로는 BLE)를 지원하는 모든 스마트기기에서 동작할 수 있기에 스마트폰과 펜만 있으면 언제 어디서든 손쉽게 필기가 가능할 것이다.

○ VR 기기

- VR 기기는 착용자의 손의 위치를 트래킹하여 디지털적으로 보여줄 수 있다. 본 팀의 작품처럼 이를 이용하면 손이 이동한 경로를 추적하고 이를 기록할 수 있지만 크기와 성능 면에서 본 팀의 작품에 비해 매우 뒤쳐진다.
- 본 팀의 작품은 IMU를 이용하여 자체적으로 경로계산에 필요한 가속도 및 자기정보 데이터를 생성해내기 때문에 별도의 트래킹 장치가 필요 없으므로 소형화할 수 있다. VR 기기의 경우 기기 자체도 크기가 큰 데다 손에 착용한 VR 기기의 위치를 트래킹하기 위해서는 별도의 장치를 방 안에 설치해야 한다.
- 본 팀의 작품은 IMU에서 보낸 가속도 및 자기정보 데이터를 처리하여 실제 경로를 계산하는 필터가 주된 프로세스로 프로세싱 파워를 많이 사용하지 않는다. VR 기기는 최소사양이 존재하는 만큼 VR 기기를 지원할 수 있는 하드웨어가 필요한데 통상적으로 이를 지원하는 하드웨어는 성능이 높은 고가의 제품인 경우가 많다. 따라서 본 팀의 작품은 전력 소모 측면이나 경제성 측면에서 VR 기기보다 우위이다.

○

-
-

□ 개발결과물의 파급력 및 기대효과

○ 모두를 위한 편리한 필기도구

- 평면에 국한되지 않은 필기를 할 수 있다는 점은 정보화 시대를 살아가고 있는 많은 사람들에게 매력적으로 느껴질 수 있는 점이다. 그 이유는 평면 즉, 태블릿이나 노트 같은 별다른 추가적인 도구를 들고 다니지 않아도 어디서나 필기를 할 수 있기 때문이다. 따라서 펜만으로 빠른 필기를 할 수 있기 때문에 청각장애인들 뿐만 아니라 그 이외의 많은 사람들에게도 충분히 수요가 있을 것이라고 판단된다.
- 청각장애인들은 ‘스페이스펜’을 이용해서 글을 통해 비장애인에게 손쉽게 정보전달이 가능하다. 목소리의 피드백 작용을 하지 못하는 청각장애인들은 정확한 발음으로 발언하기 어려워 목소리가 아닌 다른 방식으로 소통을 해왔다. 대표적인 것으로 수화와 글이다. 수화의 경우는 관련 직종이 아닌 비장애인들이 잘 모르기 때문에 일상에서 수화로 소통하기에는 어려운 점이 있다. 또한 아직 어린아이나 노인의 경우는 스마트폰의 활용능력이 부족하여 스마트폰으로 직접 타이핑하여 소통하기도 어려운 점이 있다. 따라서 이들과의 소통은 대부분 필담으로 이루어진다. 하지만 항상 펜과 종이를 들고 다니는 것은 어렵기 때문에 ‘스페이스펜’만 휴대한다면 어느 공간에서든 어느 분량의 내용이든 쉽게 즉석으로 글을 쓰고 받는 형식으로 소통을 할 수 있다. 따라서 ‘스페이스펜’은 훗날 배리어프리 사회를 만들어 나가는 핵심적인 장치로 발전 할 것이다.
- 펜으로 쓴 필기를 실시간으로 디지털 변환을 거쳐 스마트폰으로 전송된다는 점도 활용성을 높일 수 있는 부분이다. 정보의 바다라고 불리는 만큼 정보화 세상에서는 정보를 잃어버리지 않을 수 있는 형태로 지니고 있는 것이 굉장히 중요하기 때문이다. 필기를 한 즉시 스마트폰으로 전송하여 디지털 변환된 형태로 가지고 있을 수 있다는 점은 ‘스페이스펜’의 높은 판매 가치에 큰 기여를 할 수 있을 것입니다.
- 이렇듯 ‘스페이스펜’은 청각장애인의 고유한 니즈를 충족시켜줄 뿐 아니라 비장애인에게도 편리한 기능을 제공한다. 따라서 이는 결국 정보화 시대를 살아가는 우리 모두의 필기도구로 발전할 것이다.

○ 예술 행위의 진화

- 무한한 그림 공간과 정해지지 않은 평면은 예술가들에게 큰 영감을 줄 것이다. 기존의 팝아트, 그래피티들은 수직으로 세워진 정해진 평면에서만 이루어진다. 그렇기 때문에 예술가의 상상력은 한정된 도화지 위해서 펼쳐지기에 무한한 상상력을 가진 예술가에게겐 비좁을 수 있다. 스페이스펜을 사용한다면 공간이 정해져 있지 않다. 예술가의 팔이 닿은 어느 곳이든 도화지가 되며 예술가의 상상력 또한 무한히 펼칠 수 있다. 그리고 스페이스펜은 하지만 보이지 않는 무한한 평면에서 드로잉을 한다. 그렇기 때문에 이러한 특성을 이용해서 새로운 예술 문화를 만들 수 있다. 이는 가령 관객 앞에서 허공에다가 그림을 그리고 그림이 그려지는 과정을 실시간으로 보여줄수 있고 혹은 연출자의 무대가 끝난 뒤에 완성된 그림을 보여주는 등의 관객에게 예측 불가능한 결과로 부터 오는 재미를 선사 할수 있을 것이며 이는 현대 예술의 발전으로 이루어질 것이다.

□ 개발 일정

No	내용	2022年											
		7月				8月				9月			
1	IMU, Filter 조사 (경로 계산 방법 등)												
2	IMU, MCU 연결+데이터 받는 데모 (in 아두이노 나노)												
3	filter 논문분석(칼만, 상보, 매드윅 등)												
4	아두이노 블루투스 적합성 조사												
5	eval kit(실험)												
6	팀뷰어 환경 구성												
7	자기센서도 조사												
8	블루투스로 데이터 전송 1. BNO055 Bluetooth 전송 2. ABX(NANO BLE) Bluetooth 전송												
9	matlab코드 filter 수정 (Mahony-> Madgwick)												
10	프로젝션3d->2d로 구현(처음 중간 끝 점 normal vector cross product)												
11	샘플레이트하기(time interrupt) 아두이노 샘플레이트 제한												
12	bno+avr나노+시리얼 블루투스->블루투스 시리얼+pc												
13	최종 연산 구현: 정확도 문제- Madgwick 필터 적용												
14	최종 연산 구현: 실시간 처리 문제 - plotting												
15	최종 연산 구현: view point 문제 해결												
16	보고서 작성												
17	3D 프린터를 이용한 외관 제작												

□ 팀 업무 분장

No	구분	성명	참여인원의 업무 분장
1	팀장	윤지민	Trajectory 인식 논문 분석, 필터 알고리즘 분석 및 개발
2	팀원	문종찬	MATLAB 앱 작성, BLE 가상포트 개발 및 프로그램 테스트
3	팀원	박창민	Python 앱 개발, Python으로의 MATLAB 코드 이식 및 개발
4	팀원	최태웅	Arduino Nano BLE 33 블루투스 연결, Inventor 3D 모델링
5	팀원	최자은	Arduino Nano BLE 33 펌웨어 개발 및 필터 테스트