

Opensource

& GitHub

Team_init

13010529 윤웅진

14011762 이주엽

121551 서현석

Team_init

팀 장

13010529 윤웅진



Unoong

팀 원

14011762 이주엽



leejooyub

팀 원

121551 서현석



seohyeonseok

Prologue

STEP 1

주제선정 및 계획수립

STEP 3

New feature proposal

오픈소스 탐색 및 분석

STEP 2

Being Contributor

STEP 4

Step 1 - 주제 선정



Hardware : 기존 임베디드 기기에 대한 지식을 활용할 수 있는 분야

Software : 비밀번호 입력 or 카드키를 이용한 인증방식을 통한 소프트웨어적인 보안법

System : 도어락 시스템은 기기를 이용해서 공간을 물리적으로 분리시킬 수 있는 시스템으로 물리적인 보안시스템

<도어락>

결론

소프트웨어적인 보안 뿐만이 아닌 하드웨어적인 보안을 동시에 다룰 수 있는 주제를 선정하게 되었다.

Step 1 - 계획수립

Team Project 계획서

info	학과	정보보호학과	과목	오픈소스개론
	담당교수	박기웅	제출 일자	~16.11.30
과제수행형태	팀 프로젝트			
팀 명	Team init			
과 제 명	Opensource TeamProject with GIT & Git-Hub			
팀 구성원	성 명	역 할		
	윤용진	팀장 , 프로젝트 구성 및 총괄 책임		
	서현석	팀원 : 기록 및 source coding		
	이주엽	팀원 : 분석 및 source coding		
과제 개요	Git-hub를 이용해서 팀원 간의 동시 작업의 효율성을 증가시킨 프로젝트를 진행한다.			
수행 방법	1. Git-Hub에서 응용할 프로젝트(오픈 소스)를 가져 온 후 팀원 개인인 의 local repository에서 분석 및 개발 작업을 진행한다. 2. 수행 기간 동안 개선 할 점을 변경하고 지속적으로 공통 저장소 (remote repository)에 올린다.			
기대결과	git-hub와 git을 이용한 팀 프로젝트를 통해 좀 더 효율적이고 명확한 오픈소스 응용 프로젝트를 진행한다.			

1st 주제선정 및 오픈소스 검색

2nd 검색한 오픈소스 분석

3rd 분석을 통한 수정사항 토의

4th 수정사항 소스코딩 (기능 구현)

5th 구현한 기능 PULL REQUEST

Step 2 - 탐색(fork)

The image shows a GitHub search results page for the query 'doorlock arduino'. The left sidebar displays filters for Repositories (19), Code (171), Issues, Wikis (16), and Users. Below these are language filters for Arduino, C++, Java, JavaScript, and Objective-C. The main search results list several repositories, with 'Technocopia/DoorLockCode' highlighted in a red box. A red arrow points from this box to the repository's detail page on the right. The detail page for 'Technocopia / DoorLockCode' shows 18 commits, 1 branch, 0 releases, and 2 contributors. It lists files: .gitignore, 71-ti-permissions.rules, and DoorLockCode.ino. A second red arrow points from the 'Fork' button (which has a count of 2) to the 'Unoong / DoorLockCode' repository, which is marked as 'forked from Technocopia/DoorLockCode'. This forked repository shows 48 commits, 5 branches, 0 releases, and 5 contributors. It lists files: team_init, .gitignore, 71-ti-permissions.rules, and DoorLockCode.ino. The commit history for the forked repository shows a sequence of commits: 'Initial commit', 'adding udev rules file', 'comments', and 'add final annotations in the original code'.

Search: doorlock arduino

We've found 19 repository results

Technocopia / DoorLockCode

Arduino Project - NFC DoorLock

C++ Updated 10 days ago

The arduino sketch to control the RFID door lock

18 commits 1 branch 0 releases 2 contributors

Branch: master New pull request

Create new file Upload files Find file Clone or download

acamillo comments Latest commit 8965aea 21 hours ago

.gitignore Initial commit 5 months ago

71-ti-permissions.rules adding udev rules file 5 months ago

DoorLockCode.ino comments 21 hours ago

Unoong / DoorLockCode

forked from Technocopia/DoorLockCode

The arduino sketch to control the RFID door lock

48 commits 5 branches 0 releases 5 contributors

Branch: master New pull request

Create new file Upload files Find file Clone or download

This branch is 32 commits ahead, 2 commits behind Technocopia:master.

Unoong add final annotations in the original code Latest commit 85c1fcb 5 hours ago

team_init seohyeonseok : make folder a day ago

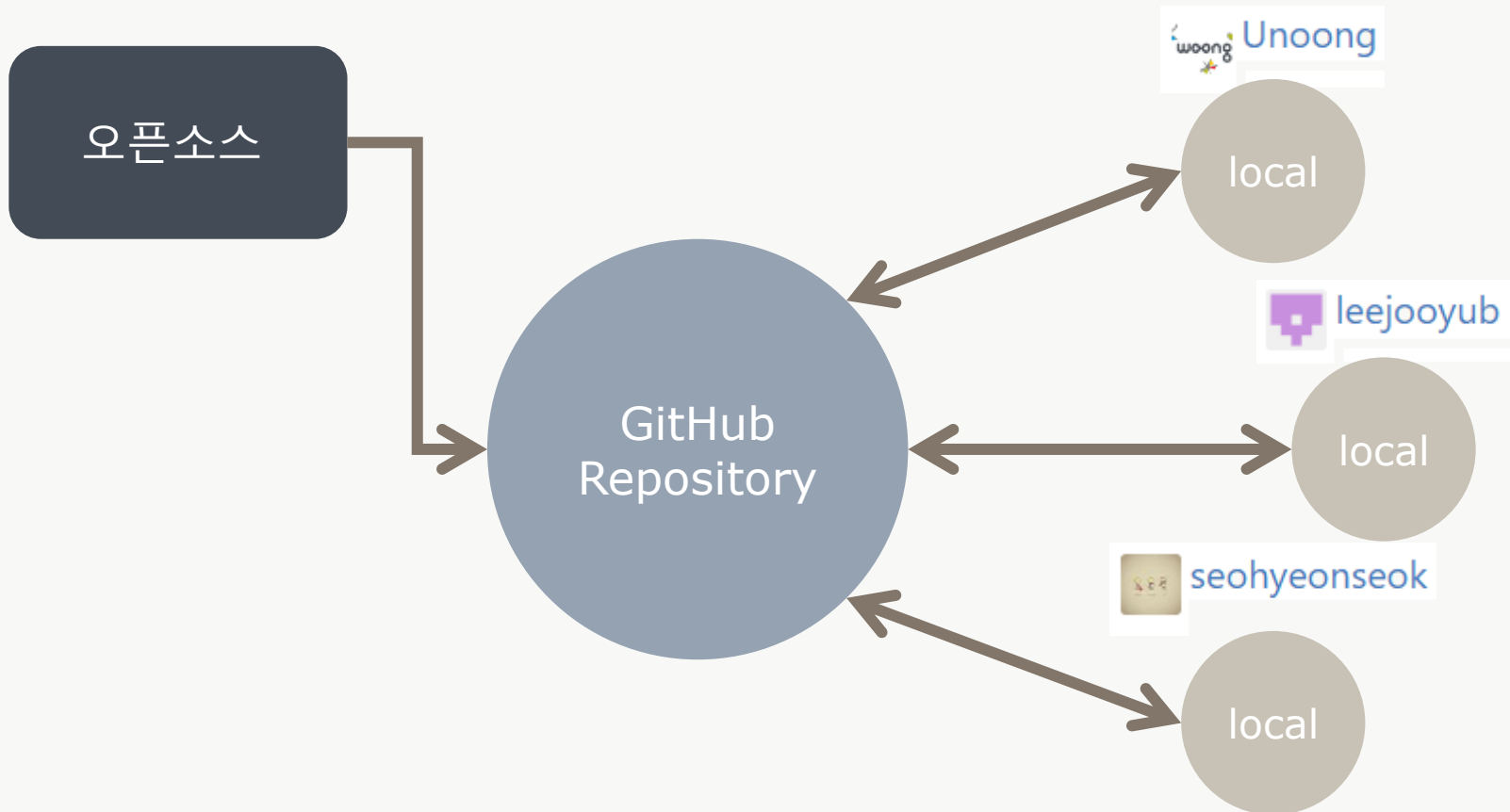
.gitignore Initial commit 5 months ago

71-ti-permissions.rules adding udev rules file 5 months ago

DoorLockCode.ino add final annotations in the original code 5 hours ago

Step 2 – git clone

```
서현석 @DESKTOP-96QDBG C MINGW64 ~/Downloads  
$ git clone http://github.com/Unoong/DoorLockCode
```



Step 2 - 분석(branch & commit)

```
서 현 석 @DESKTOP-96QDBG MINGW64 ~/Downloads
$ git branch -b "branch name"

서 현 석 @DESKTOP-96QDBG MINGW64 ~/Downloads
$ git commit -m "commit message"
```

Commits on Nov 22, 2016

- Merge branch 'master' of https://github.com/Unoong/DoorLockCode
Unoong committed 7 days ago
- modify the plan
Unoong committed 7 days ago

Commits on Nov 21, 2016

- Comment Revised
leejooyub committed 8 days ago
- modify process file
seohyeonseok committed 8 days ago
- remove push_test
leejooyub committed 8 days ago
- push_test
Joo Yub Lee committed 8 days ago
- seohyeonseok : making process file
seohyeonseok committed 8 days ago
- commented success
Joo Yub Lee committed 8 days ago

Commits on Nov 17, 2016

- seohyeonseok : modify project file
seohyeonseok committed 13 days ago

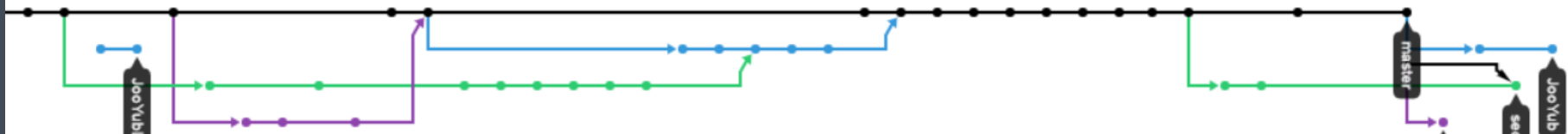
Commits on Nov 15, 2016

- seohyeonseok : making project file
seohyeonseok committed 14 days ago

Github 웹 페이지에 접속해보면 Github에서 제공하는 기능을 알 수 있는데 commit message와 Tree를 알아보기 쉽게 시각적인 자료로 제공해준다.

← commit message

↓ Tree



Step 2 - 분석(Raw & blame)

Branch: master DoorLockCode / DoorLockCode.ino

Find file Copy path

Unoong add final annotations in the original code

85c1fcb 6 hours ago

3 contributors

155 lines (124 sloc) 6.55 KB

Raw

Blame

History

Download

Share

Trash

DoorLockCode / DoorLockCode.ino

Newer Older

100644 155 lines (124 sloc) 6.55 KB

Raw Normal view History

add final annotations in the original code
Unoong authored 6 hours ago

commented success
Joo Yub Lee authored 8 days ago

Adding original code
madhephaestus authored on 3 Jul

add final annotations in the original code
Unoong authored 6 hours ago

commented success
Joo Yub Lee authored 8 days ago

Update DoorLockCode.ino
madhephaestus authored on 23 Jul

commented success
Joo Yub Lee authored 8 days ago

1 #define W0 3 //입력 단자 설정

2 #define W1 4

3 #define DoorP 5

4 #define DoorE 6

5

6 /* 기본적인 코드 형리

7 * Wiegand 프로토콜을 구현하기 위한 rfid 리더기와 자기선 리더기에 사용되는 코드이다.

8 * wiegand format이란 출입통제 시스템에 주로 사용되는 접근식 rf카드 리더에 대부분 적용되는

9 * Wiegand 프로토콜은 even parity bit, facility code, card number, odd parity bit 로

10 * 이 코드에서는 33bit의 데이터를 패리티 검사한다.

11 */

12 // 저장되어 있는 고유 card 정보를. (card number)

13

14 static unsigned long int cards[] =

15 {7142197, 7142204, 7142180, 7142175, 7142159, 7142198, 7142160, 7142242, 7142199, 7142179, 71422

16 7142177,

17 7142221,

18 7142223,

19 7142224,

20 7142225,

21 7142226,

22 7142227,

23

24

25 7142229,

26 7142230,

27 7142231,

28 7744836,

29 7744835,

30 7744834,

31 7142236,

32 7142237,

33 7142238,

34 7142239,

35 7142240,

36 7142241,

37 7142252

38

39 char bits[80];

}; // 변수를 static 으로 정의하여 유효범위를 넘어가지 않아도 소멸하지 않음

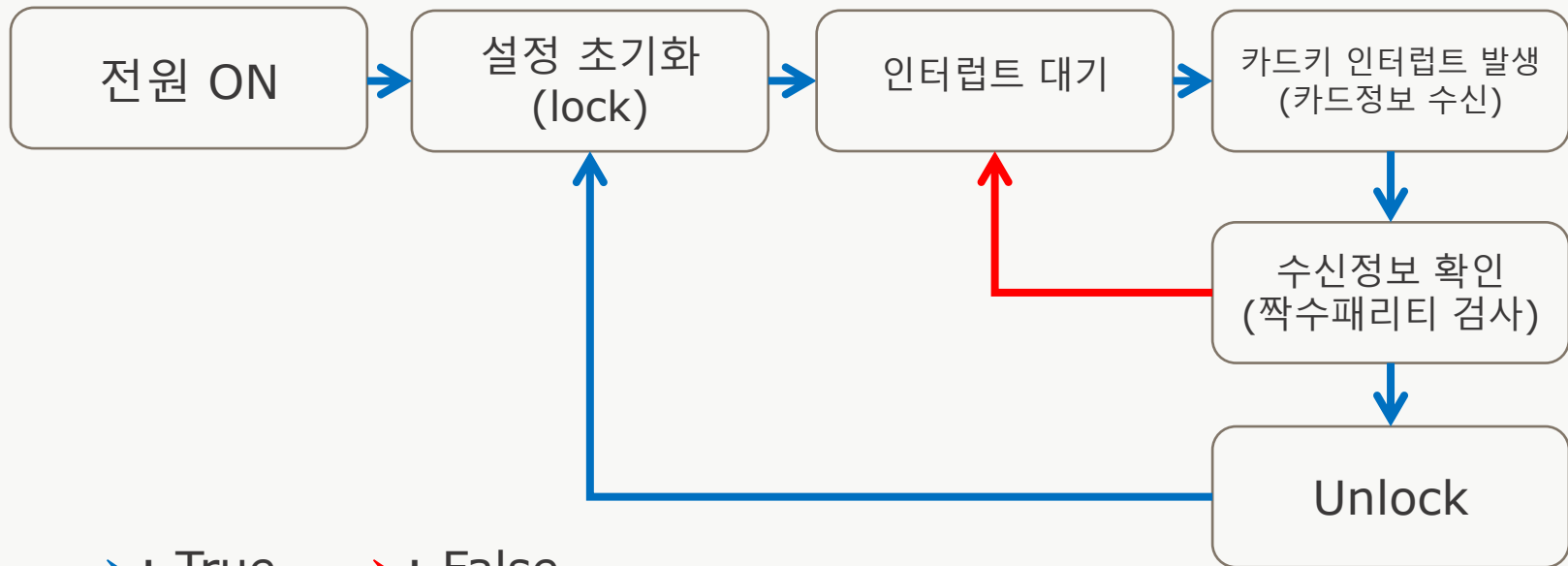
주석작업을 완료한 후 Github에서 제공하는 다른 기능을 사용해 보았다.

먼저, 소스코드에 Highlight 기능이 있어서 코드를 보기가 편하다.

Blame 버튼을 누르면 소스코드에서 추가된 부분과 추가한 사람에 대한 정보를 보여주는데 추가한 시간 별로 색으로 표시해 준다.

Raw 버튼은 한마디로 정말 날 것의 텍스트만 존재하는 페이지로 연결시켜 준다.

Step 2 - 분석(algorithm)



→ : True → : False

bitw : 카드정보 [ep(1bit)+facility(7bit)+card(24bit)+op(1bit)]

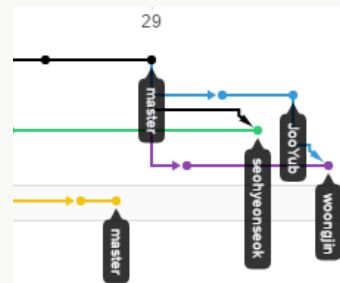
timeout : 대기시간을 설정

ep : even parity bit

op : odd parity bit

site : facility code

card : card number



※ 해석도중에 저자가 원래 코드에 자세한 주석을 추가해 주었습니다.

주황색 branch가 저자의 branch

Step 3 – New feature proposal

[소스코드의 기능]

소스코드는 카드키 알고리즘을 이용한 도어락 코드

[추가 옵션]

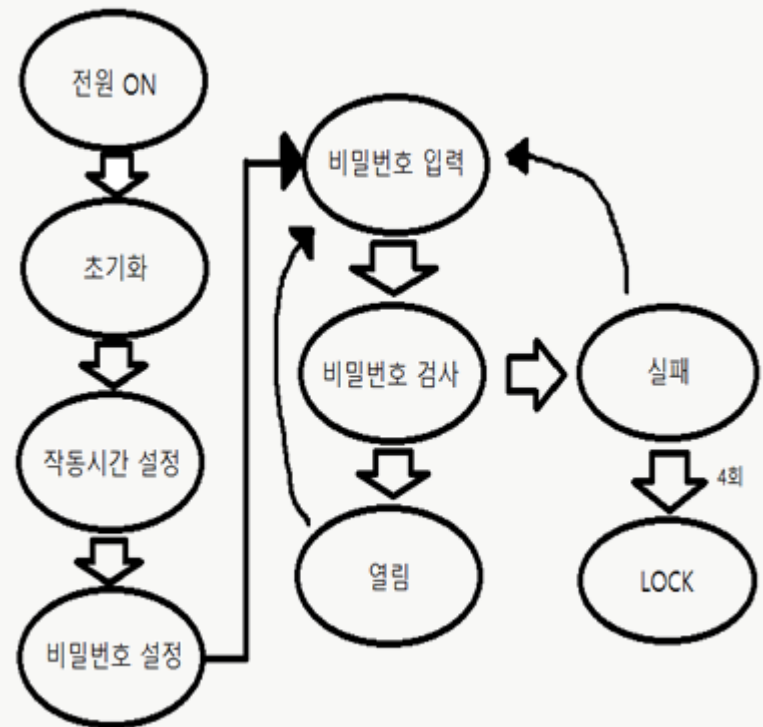
- 카드키 알고리즘 이외의 비밀번호를 이용한 알고리즘을 추가
- 기기의 작동시간을 지정해서 작동시간 이외의 인터럽트는 무시(사용시간 설정)
- 4회 이상 잘못된 입력이 있으면 기기가 작동을 멈춤

[역할 분담]

윤웅진 - 도어락의 사용시간을 설정하는 기능

이주엽 - 키패드를 이용할 수 있는 기능, 오류횟수에 따른 작동정지

서현석 - 비밀번호를 대칭키를 이용해서 암호화



Step 4 – 구현(push & pull & merge)

```
서현석 @DESKTOP-96QDBG MINGW64 ~/Downloads
$ git pull "remote name"

서현석 @DESKTOP-96QDBG MINGW64 ~/Downloads
$ git merge "branch name"

서현석 @DESKTOP-96QDBG MINGW64 ~/Downloads
$ git push "remote name" "branch name"
```

```
const byte rows = 4;
const byte cols = 4;
// 키패드의 행, 열의 갯수

char keys[rows][cols] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
// 키패드 버튼 위치 설정

byte rowPins[rows] = {8, 7, 6, 5};
byte colPins[cols] = {4, 3, 2, 1};
// 키패드에 연결된 핀번호 설정 (데이터 시트 참고)

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);
// 키패드 오브젝트 생성

int redPin = 13;
int greenPin = 12;
// LED 2개의 핀번호 설정
```

키패드와 비밀번호입력 및 암호화는 구현할 수 있었지만
사용시간을 설정하는 부분은 구현하지 못하였다.

```
void makepassword() {
  while(1) {
    temp = keypad.getKey();
    if(temp == '1') {
      password = password*10+1;
    }
    if(temp == '2') {
      password = password*10+2;
    }
    if(temp == '3') {
      password = password*10+3;
    }
    if(temp == '4') {
      password = password*10+4;
    }
    if(temp == '5') {
      password = password*10+5;
    }
    if(temp == '6') {
      password = password*10+6;
    }
    if(temp == '7') {
      password = password*10+7;
    }
    if(temp == '8') {
      password = password*10+8;
    }
    if(temp == '9') {
      password = password*10+9;
    }
    if(temp == '0') {
      password = password*10+0;
    }
    if(temp == '*' || temp == '#') {
      password = password ^ key_security;
      break;
    }
  }
}

void matchpassword()
{
  while(1) {
    temp = keypad.getKey();
    if(temp == '1') {
      password_input = password_input*10+1;
    }
    if(temp == '2') {
      password_input = password_input*10+2;
    }
    if(temp == '3') {
      password_input = password_input*10+3;
    }
    if(temp == '4') {
      password_input = password_input*10+4;
    }
    if(temp == '5') {
      password_input = password_input*10+5;
    }
    if(temp == '6') {
      password_input = password_input*10+6;
    }
    if(temp == '7') {
      password_input = password_input*10+7;
    }
    if(temp == '8') {
      password_input = password_input*10+8;
    }
    if(temp == '9') {
      password_input = password_input*10+9;
    }
    if(temp == '0') {
      password_input = password_input*10+0;
    }
    if(temp == '*' || temp == '#') {
      break;
    }
    if(password_input == (password ^ key_security)) {
      digitalWrite(DoorP, HIGH); // DoorLock 이 열린다.
      delay(3000);
    }
  }
}
```

Step 4 – Pull request

The screenshot shows the GitHub repository page for **Technocopia / DoorLockCode**. The repository description is "The arduino sketch to control the RFID door lock". It has 18 commits, 1 branch, 0 releases, and 2 contributors. A red box highlights the **New pull request** button. A red arrow points from this button to the text **Open a pull request**, which includes the instruction: "Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#)."

Below this, the pull request interface is shown. The base fork is **Technocopia/DoorLockCode**, the base branch is **master**, and the head fork is **Unoong/DoorLockCode** with the compare branch **master**. A message states: "Can't automatically merge. Don't worry, you can still create the pull request."

The pull request title is **Add option in DoorLockCode**. The description contains the text: "Using keypad, making the password. it makes doorlock open." A red box highlights the **Create pull request** button at the bottom right. A red arrow points from this button to the text **Add option in DoorLockCode #1**, which is the pull request title. Below the title, a list of commits is shown, including "seohyeonseok : making project file", "seohyeonseok : modify project file", "commented success", "seohyeonseok : making process file", "push_test", "remove push_test", "modify process file", "Comment Revised", "modify the plan", "Merge branch 'master' of https://github.com/Unoong/DoorLockCode", "seohyeonseok : encryption", "seohyeonseok : add matchpassword", "seohyeonseok : modify makepassword.c", "seohyeonseok : modify makepassword.c", "seohyeonseok : modify makepassword.c", "seohyeonseok : before merge", "seohyeonseok : before commit merging seohyeonseok", "seohyeonseok : before commit merging seohyeonseok", and "seohyeonseok : after merge".

Epilogue - 활동 로그

[process of teamwork]

11/11 - first meet

NOTE

주제 정하기

다음 일정 조정

git 숙달

11/15 - second meet

NOTE

주제에 대한 세부내용 선정

계획서 작성

과제 수행에 앞서 자료수집

github에 있는 repository에
fork

수정할 파트 선정

11/21 - third meet

NOTE

공부한 내용 공유

자료 정리 및 알고리즘 만들기

수정할 역할 분담

11/25 - fourth meet

NOTE

-C 언어로 알고리즘 구현

-일정 부분 ino 코드에 적용하기

11/28 - fifth meet

NOTE

-hwp문서 합치기

-ino 코드 완성 & 주석 작업

-발표 구상

11/29 - sixth meet

NOTE

-코드 수정 작업

-발표자료 완성

과제를 수행하면서 중간과정이 계속 기록이 되는 것을 보면서 '협업을 하고 있구나' 라는 느낌을 받을 수 있었고, 저자의 branch가 생성되서 자세한 주석이 달리는 것을 보면서 직접 협업 토의를 거치지 않고 이렇게 Github를 통해서 협업을 하는 새로운 경험도 하게 되었다.

Epilogue

Git에서 사용한 기능

- clone
- branch
- checkout
- commit
- push
- pull
- merge

Github에서 이용한 서비스

- fork
- code view
- raw
- blame
- commit view
- tree view
- pull request

사용하지 않은 기능

Wiki

Github에서 제공하는 서비스에는 wiki기능이 있다. Wiki기능은 사용자가 직접 repository에 대한 설명을 할 때 사용할 수 있는 기능이다. 문서 형태로 편집이 가능하다. 이번 과제에는 발표자료를 별도로 만들었기 때문에 이용하지 않았다.

Tag

Repository에 이름을 붙여서 version을 만들어 주는 기능이다. Github 웹 페이지는 repository임과 동시에 이러한 관리에 관한 기능도 제공해준다.

Q&A

