

후회하지말고 최선을 다해서

Search...

Embedded System/ATmega128

[ATmega128] DFPlayer_Mini

전두선 2019. 8. 16. 14:33

DFPlayer_Mini

DFPlayer+Mini+Manual.pdf
0.41MB

FN-M16P+Embedded+MP3+Audio+Mod...
0.79MB

- 이 오디오 코덱 모듈은 Micro SD카드에 있는 MP3 파일(또는 WMV)을 재생시켜주는 모듈입니다.


디바이스 마트	9,900	http://www.devicemart.co.kr/1278727
aliexpress	5pcs, US \$5.59(6,384)	

후회하지말고 ... 구독하기

https://fr.aliexpress.com/item/5pcs-DFPlayer-Mini-MP3-Player-Module-MP3-Voice-Module-for-Arduino-DIY-Supporting-TF-Card-and/32769724485.html?spm=2114.01010208.3.1.i6qdJo&ws_ab_test=searchweb0_0%2Csearchweb201602_2_10152_10065_10151_10068_10136_10137_10060

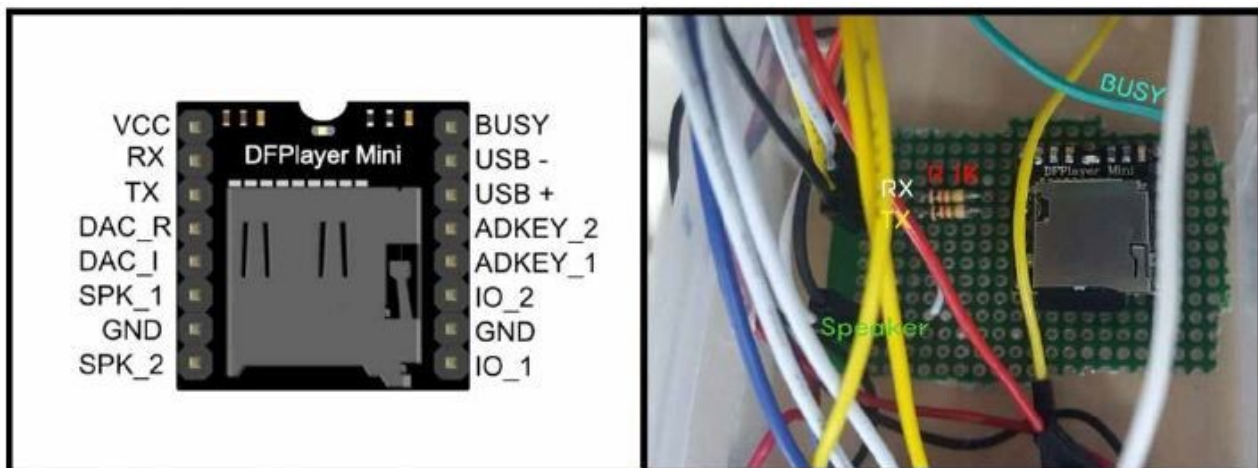
국내와 해외 가격이 많이 차이니까 구매하실때 참고하세요!

그리고 모듈만 있어서는 안되고 MP3 파일을 담을 Micro SD도 같이 가지고 계셔야 합니다.



- MP3 및 WMV 디코딩 지원
- 지원 샘플링 속도
(8KHz, 11.025KHz, 12KHz, 16KHz, 22.05KHz, 24KHz, 32KHz, 44.1KHz, 48KHz)
- FAT16, FAT32 파일 시스템, 최대 32GB TF 카드 지원
- 직렬 통신(9600BAUD), AD Key 등 다양한 제어 방법
- 30 레벨 볼륨 조절 가능, 10 레벨 EQ 조절 가능.
- DC 3.2V ~ 5.0V
- 20 mA

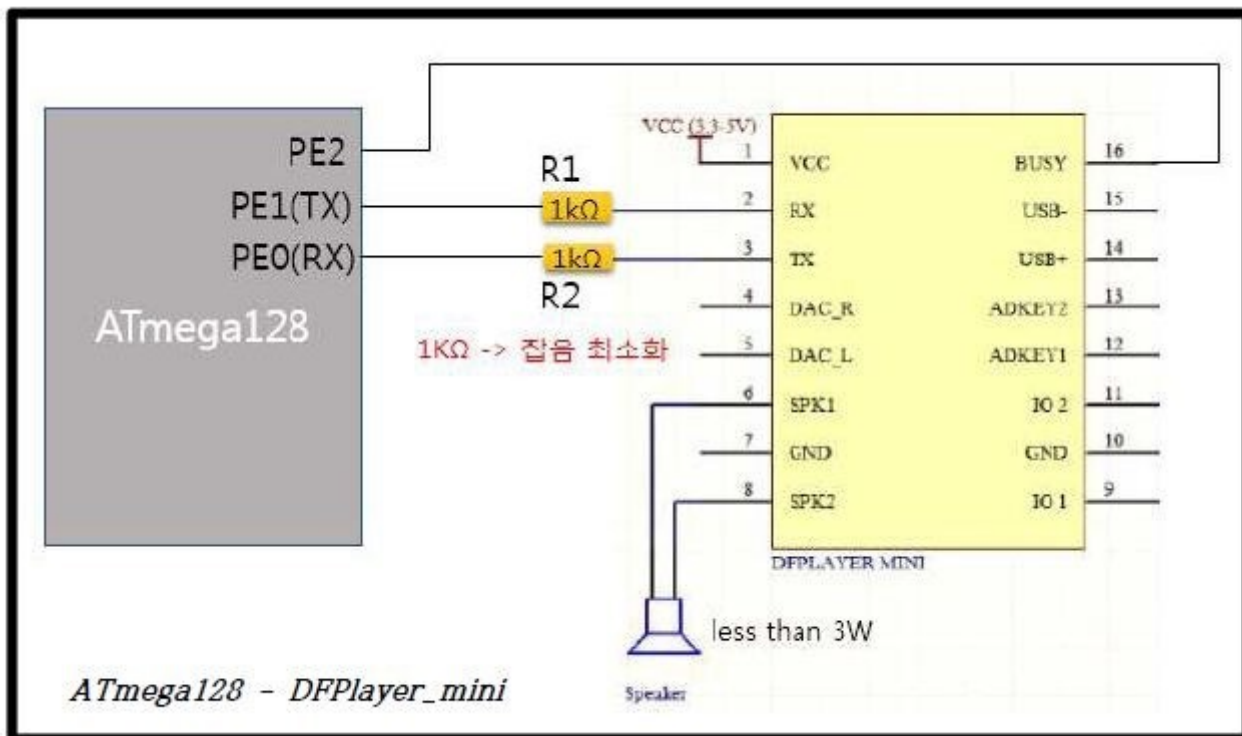
<DFPlayer>



/* 5V 인가시, Tx, Rx에 필히 1k 저항을 달아주셔야 잡음이 안난다는걸 주의하세요! */

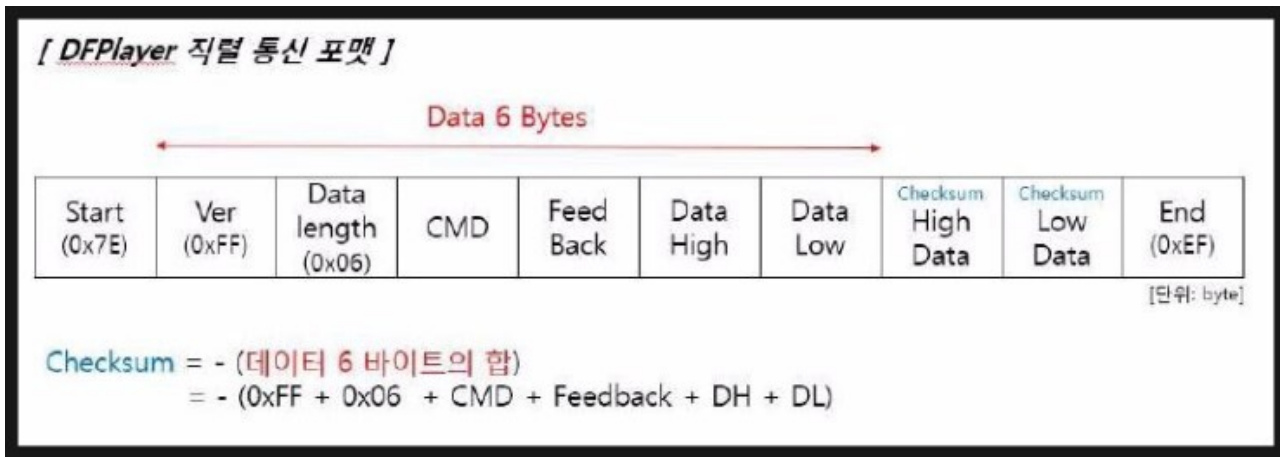
후회하지말고 ... 구독하기

핀번호	이름	설명	비고
1	VCC	입력전원	3.2V~5V, 일반적으로 4.2V
2	RX	UART 시리얼 입력	5V 인가 시, 1K 저항을 직렬로 연결(잡음 영향 최소화)
3	TX	UART 시리얼 출력	5V 인가 시, 1K 저항을 직렬로 연결(잡음 영향 최소화)
4,5	DAC_R,L	오디오 출력	이어폰 연결
6,8	SPK_1,2	스피커	3W 미만의 스피커 연결
7,10	GND	접지	파워 접지
9,11	IO_1,2	트리거 포트	짧게 누르면 이전/다음곡 플레이, 길게 누르면 볼륨 작게/크게
12,13	ADKEY_1,2	AD포트	첫 번째 세그먼트 플레이(1), 다섯 번째 세그먼트 플레이(2)
14,15	USB+ USB-	USB+ DP USB- DM	USB 포트
16	BUSY	플레이 상태	Low: 플레이 중 High: 플레이 정지상태



저는 UART 통신으로 제어하는걸 원해서 이렇게 사용했지만, DFPlayer는 그밖에도 ADKey, IO 모드를 제공하니 다른 모드를 원하시는분은 데이터 시트를 참고하세요.

<UART 포맷 : 9600/8/n/1/n>



시리얼 통신으로 DFPlayer를 제어하기 위해서 데이터 시트에서 요구되는 조건을 충족시켜야 제어가 가능한데,

UART 통신의 포맷설정은 당연히 기본이고, 위 사진의 **직렬 통신 포맷** 순서도 지켜줘야 합니다.

간단하게 설명하면 명령의 시작을 알리는 Start Byte를 시작으로 사용자가 원하는 명령에 대한 데이터 6 Byte를 보내고 그에 따른 데이터 6 Byte의 Checksum을 계산 후 보내고 명령의 끝을 알리는 End Byte를 보내면 통신은 마무리 됩니다.

명령에 대한 정보는 데이터 시트에 자세하게 나와 있으므로 참고 바랍니다!

예를 들어, SD Card에 저장된 0001.mp3를 재생하려면 0x7E(Start)/FF/06/12(Play CMD)/00/00/01(0001.mp3 Play)/FE/E8/EF(End) 가 됩니다. 추가적으로 Checksum = -(FF+06+12+00+00+01) = 0xFEE8이 되기 때문에, Checksum high byte는 0xFE, 그리고 Low byte는 0xE8이 됩니다.

MP3와 WMV에 대한 하드웨어 디코딩을 지원하며, FAT16과 FAT32 파티션 지원으로 최대 32GB까지의 메모리를 인식합니다.

- 노래는 반드시 SD 카드의 최상위 폴더에 mp3 폴더를 만들어서 mp3 폴더 안에 넣어야 합니다.
 그리고 mp3 파일명을 **0001 ~ 9999** 까지 번호를 원래의 MP3 파일명의 앞에 붙여주면 됩니다.
 또는 번호만 네이밍 해도 문제없습니다.
(ex> 0001.mp3 or 0001music.mp3)

```

1  /  ATmega128 0.2  /
2  /* ATmega128 */
3  #define F_CPU 8000000UL
4  #include <avr/io.h>
5  #include <util/delay.h>
6  //
7  typedef unsigned char INT8;
8  typedef unsigned int INT16;
9  //
10 #define BAUD 9600
11 #define U2X_S 2 // Set of U2X --> 1 or 2
12 #define MYUBRR ((F_CPU*U2X_S)/(16L*BAUD)-1)
13 #define sbi(reg,bit) reg |= (1<<(bit)) // Set "bit"th bit of reg
14 #define cbi(reg,bit) reg &= ~(1<<(bit))
15 //
16 #define MP3_NEXT 0x01
17 #define MP3_PREVIOUS 0x02
18 #define MP3_TRACKING_NUM 0x03 // 0..2999
19 #define MP3_INC_VOLUME 0x04
20 #define MP3_DEC_VOLUME 0x05
21 #define MP3_VOLUME 0x06 // 0..30
22 #define MP3_EQ 0x07 // 0-Normal / 1-Pop / 2-Rock
23 #define MP3_PLAYBACK_MODE 0x08 // 0-Repeat / 1-folder repeat
24 #define MP3_PLAYBACK_SOURCE 0x09 // 0-U / 1-TF / 2-AUX / 3-USB
25 #define MP3_STANDBY 0x0A
26 #define MP3_NORMAL_WORK 0x0B
27 #define MP3_RESET 0x0C
28 #define MP3_PLAYBACK 0x0D
29 #define MP3_PAUSE 0x0E
30 #define MP3_PLAY_FOLDER_FILE 0x0F // 0..10
31 #define MP3_VOLUME_ADJUST 0x10
32 #define MP3_REPEAT 0x11 // 0-stop play / 1-start
33 // Query the System Parameters
34 #define MP3_Q_STAY1 0x3C
35 #define MP3_Q_STAY2 0x3D
36 #define MP3_Q_STAY3 0x3E
37 #define MP3_Q_SEND_PRM 0x3F
38 #define MP3_Q_ERROR 0x40
39 #define MP3_Q_REPLY 0x41
40 #define MP3_Q_STATUS 0x42
41 #define MP3_Q_VALUE 0x43
42 #define MP3_Q_EQ 0x44
43 #define MP3_Q_PLAYBACK_MODE 0x45
44 #define MP3_Q_SOFT_VERSION 0x46
45 #define MP3_Q_TF_CARD_FILES 0x47
46 #define MP3_Q_U_DISK_CARD_FILES 0x48
47 #define MP3_Q_FLASH_CARD_FILES 0x49
48 #define MP3_Q_KEEPPON 0x4A
49 #define MP3_Q_CURRENT_TRACK_TF 0x4B
50 #define MP3_Q_CURRENT_TRACK_U_DISK 0x4C
51 #define MP3_Q_CURRENT_TRACK_FLASH 0x4D

```

후회하지말고 ... 구독하기

```

52 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////,
53 //Commands parameters
54 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////,
55 #define MP3_EQ_Normal 0
56 #define MP3_EQ_Pop 1
57 #define MP3_EQ_Rock 2
58 #define MP3_EQ_Jazz 3
59 #define MP3_EQ_Classic 4
60 #define MP3_EQ_Base 5
61 #define MP3_PLAYBACK_MODE_Repeat 0
62 #define MP3_PLAYBACK_MODE_folder_repeat 1
63 #define MP3_PLAYBACK_MODE_single_repeat 2
64 #define MP3_PLAYBACK_MODE_random 3
65 #define MP3_PLAYBACK_SOURCE_U 0
66 #define MP3_PLAYBACK_SOURCE_TF 1
67 #define MP3_PLAYBACK_SOURCE_AUX 2
68 #define MP3_PLAYBACK_SOURCE_SLEEP 3
69 #define MP3_PLAYBACK_SOURCE_FLASH 4
70
71 INT8 default_buffer[10] = {0x7E, 0xFF, 0x06, 0x00, 0x00, 0x00, 0x00,
72 volatile INT8 mp3_cmd_buf[10] = {0x7E, 0xFF, 0x06, 0x00, 0x00, 0x00,
73
74 void USART0_Init( INT16 ubrr )
75 {
76     // Set baud rate
77     UBRR0H = (INT8)(ubrr>>8);
78     UBRR0L = (INT8)ubrr;
79     // Enable U2X
80     if(U2X_S == 2)
81         sbi(UCSR0A, U2X);
82     else
83         cbi(UCSR0A, U2X);
84     // Enable receiver and transmitter
85     sbi(UCSR0B, RXEN);
86     sbi(UCSR0B, TXEN);
87     // Set frame format: 8data, 1stop bit
88     cbi(UCSR0C, UMSEL); // asynch
89     cbi(UCSR0C, USBS); // 1 Stop bit
90     cbi(UCSR0C, UPM01); // No parity
91     cbi(UCSR0C, UPM00);
92     cbi(UCSR0B, UCSZ02); // 8-bit
93     sbi(UCSR0C, UCSZ01);
94     sbi(UCSR0C, UCSZ00);
95 }
96 void USART0_Transmit( char data )
97 {
98     // Wait for empty transmit buffer
99     while ( !( UCSR0A & 0x20 ) ) // (1<<UDRE)
100 ;
101 // Put data into buffer, sends the data

```

후회하지말고 ... 구독하기

```

102     UDR0 = data;
103 }
104
105 INT16 MP3_checksum (void)
106 {
107     INT16 sum = 0;
108     INT8 i;
109     for (i=1; i<7; i++) {
110         sum += mp3_cmd_buf[i];
111     }
112     return -sum;
113 }
114 void MP3_send_cmd (INT8 cmd, INT16 high_arg, INT16 low_arg)
115 {
116     INT8 i;
117     INT16 checksum;
118     mp3_cmd_buf[3] = cmd;
119     mp3_cmd_buf[5] = high_arg;
120     mp3_cmd_buf[6] = low_arg;
121     checksum = MP3_checksum();
122     mp3_cmd_buf[7] = (INT8) ((checksum >> 8) & 0x00FF);
123     mp3_cmd_buf[8] = (INT16) (checksum & 0x00FF);
124     for( i=0; i< 10; i++){
125         USART0_Transmit(mp3_cmd_buf[i]);
126         //putchar(mp3_cmd_buf[i]);
127         mp3_cmd_buf[i] = default_buffer[i];
128     }
129 }
130 void dfplayer_init(void)
131 {
132     MP3_send_cmd(MP3_PLAYBACK_SOURCE,0,MP3_PLAYBACK_SOURCE_TF);_del:
133     MP3_send_cmd(MP3_VOLUME, 0, 30); _delay_ms(10);
134 }
135
136 int main(void)
137 {
138
139     USART0_Init(MYUBRR); // 9600/8/n/1/n
140
141     dfplayer_init();
142
143     while(1)
144     {
145         MP3_send_cmd(0x12,0,1); _delay_ms(5000); // 0001.mp3 Play
146     }
147 }
148

```


BUSY 체크 핀을 이용해서 최소 딜레이만 주고 MP3 재생도 가능합니다. (데이터 시트 참고)



ATmega128 – DFPlayer_Mini
Register 871 0

00:00 00:08 1080p

2

구독하기

Tag

ATMEGA128

Avr

DFPlayer_mini

'Embedded System/ATmega128'의 다른글

이전글 [백준 1759, c++] 암호 만들기

현재글 : [ATmega128] DFPlayer_Mini

다음글 프로세싱을 이용한 시리얼 데이터 실시간 그래프화

관련글

후회하지말고 ... 구독하기

[ATmega128] sg90(서보모터)

[ATmega128] Smart Moving

HC-SR04(초음파 센서)의 이해

게시

2019.10.02

ATmega128의 펌웨어

2019.09.18

2017.08.13



댓글 15

도현 2019.12.12 15:25

보드는 어떤 것을 쓰셨나요? j kit 으로 mini mp3 모듈을 써보려 하는데, 사용하
신 보드 모양이 달라서, 궁금하네요.

ㅎㅎ

답글

전두선 2019.12.13 20:34 신고

혹시 JKIT-128 모델을 쓰신다면 저랑 같은 ATmega128의 8bit CPU를
사용하신 거예요. 영상에 나온 보드는 학교에서 교수님께서 따로 보드를
설계하셔서 만든거라 모양이 다르지만 사용방법은 같아요. UART 통신
포트 PE1과 PE0에 잘 연결하셔서 따라해보세요.

미니언즈 2019.12.18 19:20

안녕하세요! 게시글대로 따라 했는데 음악이 재생되는건 아니고 지지직 거리는 소
리만 나는 상황입니다..ㅠㅠ

혹시 사용하신 스피커 정보 좀 알려주실 수 있나요??

저희는 1W, 8ohm 짜리 초소형 미니 스피커를 이용했는데, 1kilo ohm이 너무 커
서 소리가 안 나는 건가 하고 의심중이라서...ㅠㅠ

답글

전두선 2019.12.18 20:02 신고

데이터 시트상에는 3W미만의 스피커를 사용하라고 나와있습니다. 제가
사용했던건 1~2W 짜리였는데 좀 오래되서 정확한 제품명은 기억이 안
나네요. 작성자님 말씀대로 스피커쪽 문제일 수도 있지만 시스템 클럭이
나 연결부분을 다시한번 살펴보세요! 해당 글은 ATmega128/8MHz/U
ART0 사용중입니다.

미니언즈 2019.12.20 22:25

16MHz로 바꾸니까 성공했네요! 좋은 게시글 감사합니다!! 정말 복 받으실 거예요

ㅎㅎ

답글

후회하지말고 ... 구독하기



전두선 2019.12.21 13:48 신고
다행이네요! ㅎㅎ 미니언즈님도 복 많이 받으세요~~

도현 2019.12.21 23:48

답변 감사합니다.
지금 sd 카드를 넣어서 통신하고 있는데,
default 설정으로 홈 디렉토리에 있는 노래1 만 불러고 있는 상황입니다.

위의 코드로 해봤었는데 혹시 해결방법을 아시는지요

답글

전두선 2019.12.22 17:03 신고

데이터 시트 내부에 보시면 Specify playback of folder named "MP3"의 명령어가 0x12라고 나와있고, 현재 0001.mp3를 재생시키기 위해 MP3_send_cmd(0x12,0,1); 명령어를 통해 CMD -> 0x12, DATA High -> 0x00, DATA Low -> 0x01을 넣어준 상태입니다. 도현님께서 다른 노래를 재생시키고자 하시면 DATA 쪽을 건드리시면 됩니다.

MP3_send_cmd(0x12,0,1); 은 내가 "MP3에 폴더 내부에 있는 0001번의 mp3를 재생시키겠다"라는 뜻입니다.

MP3_send_cmd(0x12,0,2); // 0002.mp3가 재생됩니다.

DF_Player는 데이터 시트를 보시면 알겠지만 기능들이 정말 많습니다. 제가 올린 건 그중에서 그냥 재생부분만 올린거구요. 다양한 기능들이 있으니 데이터 시트 참고하시면 더 많은 기능들을 이용하실 수 있을거예요!

답글

익명 2020.04.21 20:02

비밀댓글입니다

답글

임격정 2020.07.16 09:15

안녕하세요 올려주신 코드로 atmega8을 이용하여 UART통신을 하려고하는데요. 스피커를 모듈에 바로 연결하지 않고 DAC에 꽂아 AUX선으로 앰프를 통해 사용하려고 하는데 이경우에는 명령어를 어떻게 주어야하나요?

답글

후회하지말고 ... 구독하기

전두선 2020.07.16 15:19 신고

저도 DAC를 사용해서 해보진 않아서 정확한 답변을 드릴 수가 없네요.
데이터 시트를 참고해보시고, 아마 위 코드와 같은 명령으로 해도 문제
없이 동작하지 않을까 싶네요.



조동아리 2021.06.21 15:09

올려 주신 코드로 ATmega128A에 같은 mp3모듈 사용해봤는데 소리가 안나네요
ㅌㅌ

3W짜리 스피커를 사용해서 그런건지 원인을 잘 모르겠네요 ㅠㅠ

혹시 3W짜리 스피커를 사용하면 아예 소리가 안나나요??

도움 부탁드립니다 ㅌㅌ

답글

전두선 2021.06.21 15:14 신고

스피커보다는 H/W 연결부분이나 sd카드안에 mp3/0001.mp 형태로
구성하셨는지 체크해보시는게 좋을듯싶네요. 해당 코드는 문제없이 동
작한 코드여서, 소스코드보다는 외적인 요소부분을 먼저 체크해보세요

ㄷㅈㅎㄹㅈ 2021.09.17 02:55

처음 설정할때 dfplayer_init() 문에서

```
MP3_send_cmd(MP3_PLAYBACK_SOURCE,0,MP3_PLAYBACK_SOURCE_T  
F);_delay_ms(10);
```

0x09 와 1을 보내셨던데 1은 USB 고 2가 SD 카드던데 1로 주셔도 되던가요?

답글

s00o 2022.02.02 14:17

관리자의 승인을 기다리고 있는 댓글입니다

답글

이름

비밀번호

후회하지말고 ... 구독하기

내용을 입력해주세요.



☐ 비밀글

등록

< 1 ... 127 128 129 130 131 132 133 134 135 ... 139 >

더보기



jeonds1127@korea.ac.kr

분류 전체보기 (139)

Interview (1)

사회이슈 (6)

자율주행 (2)

프로그래머스 (14)

백준 (54)

알고리즘 이론정리 (4)

DP (14)

DFS (20)

BFS (13)

브루트 포스 (2)

시뮬레이션 (1)

Deep Learning (4)

OS & Tools (18)

Linux (6)

ROS (4)

후회하지말고 ... 구독하기

Unreal Engine 4 (2)

Error (6)

Sensor (12)

LIDAR (3)

mmWave Radar (4)

Camera (5)

Embedded System (22)

ATmega128 (6)

STM32F103 (2)

Raspberrypi (5)

Arduino (2)

유틸리티 (7)

기타 (5)

Tag

object detection, python3, C++, topdown, dfs,
bottomup, Homography, 다리를 지나는 트럭,
deque, BOJ 9095, 백준, Bottom up, Queue,
프로그래머스, dp, BFS, dynamic programming,
자율주행, IWR6843ISK, raspberry pi4

최근글 인기글

[Lenovo] Thinkpad t480 노트북 전원 안 켜질때 해결
방법

2021.08.07 23:24

[OS] 프로세스 vs 스레드, 뮤텍스와 세
마포어

2021.07.21 13:07

[프로그래머스, c++] 튜플

2021.06.27 17:59

최근댓글

[승인대기]

s00o

[승인대기]

후회하지말고 ... 구독하기

구매자

엄청난 정리 감사합니다!
성빈이



공지사항

Facebook Twitter

Archives

- 2021/08
- 2021/07
- 2021/06
- 2021/05

〈 2022. 04 〉						
일	월	화	수	목	금	토
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Total
115,688

Today : 40
Yesterday : 161

검색내용을 입력하세요.

후회하지말고 ... 구독하기



Copyright © Kakao Corp. All rights reserved.

관련사이트

후회하지말고 ... 구독하기