0. 용어 정리

AVR = 마이크로 프로세서

마이크로 프로세서 = 단일 칩으로 구성되어 입력된 프로그램에 의해 데이터를 입력받아 처리하고 결과를 출력하는 장치

마이크로 컨트롤러 = 마이크로 프로세서중에서 자동화기기를 제어하기에 최적으로 구현되어있는 프로세서

=================================================================================

1. 왜 굳이 비쥬얼 스튜디오 말고 아두이노를 사용하는지, 다른거 사용해도 되는지

당연히 코딩 숙련도가 높으면 비주얼 스튜디오나 파이썬같은게 나음

아두이노를 쓰는 이유는 그게 비교적 쉬워서

실제로 아두이노 라이브러리는 비주얼스튜디오같은 다른 코딩프로그램으로 만드는 경우가 많음

AVR 기반의 마이크로컨트롤러여서 다른 프로그램으로 코드 짜도 아두이노 설정만 맞추면 업로드 가능

게임을 비쥬얼스튜디오로 개발할 수 있으나 엔진으로 개발하는 것과 비슷한원리

link = http://blog.naver.com/PostView.nhn?blogId=roboholic84&logNo=220815649199

=================================================================================

2. 아두이노가 뭔지, 아두이노 스케치가 뭔지 좀 더 구체적으로

아두이노= 물리적인 세계를 감지하고 제어할 수 있는 인터랙티브 객체들과 디지털 장치를 만들기 위한 도구로

간단한 마이크로컨트롤러 보드를 기반으로 한 오픈 소스 컴퓨팅 플랫폼과 소프트웨어 개발환경.

2005년, 이탈리아 Ivrea에서 초기 프로젝트가 만들어졌다.

2008년 10월, Arduino Duemilanove가 출시되었다

2009년 3월, Arduinio Mega가 출시되었다.[14]

2011년 3월, 전 세계적으로 30만 개 이상의 아두이노 유닛을 사용할 수 있다.[15][16]

2015년 3월, 아두이노가 10주년을 맞았다.

상표권 분쟁으로 인해 2015년 5월 Arduino LLC에 의해 "Genuino"란 상표가 등록 되었고

미국 이외에 시장에서 Arduino LLC의 브랜드 이름이다.

아두이노 스케치 = 아두이노 자체가 idil에서 만든 마이크로컨트롤러

요약 효율적으로 소프트웨어를 개발하기 위한 통합개발환경 소프트웨어 어플리케이션 인터페이스이다.

코드 편집기, 디버거, 컴파일러, 인터프리터 등을 포함하고 개발자에게 제공한다.

Link = <https://www.arduino.cc/en/Main/FAQ#toc2>

=================================================================================

3. 왜 아두이노 스케치가 아니라 아두이노IDE인지

IDE

[ Integrated Development Environment ]

=================================================================================

4. 이두이노 경로 변경 가능한지, 바꾸면 어떻게 되는지&

5. INO파일이 뭔지, 텍스트로 만들어서 ino로 말만 변경해도 실행되는지

경로 변경 가능. 바꾼다면 해당 파일이 존재하는 위치에 비어있는 라이브러리를 만들고 실행된다.

아두이노에서 만든 소프트웨어 프로그램. 아두이노 프로그래밍 언어로 작성된 소스코드가 포함되어있다.

아두이노 회로 기판을 제어하는 데 사용

INO = "Arduino"의 마지막 세 문자에서 파생 됨)을 저장합니다

텍스트 파일로 파일을 만들면 파일실행가능

LINK = https://how2open.com/extension/ino

=================================================================================

6. Arduino.h 안의 main문 분석

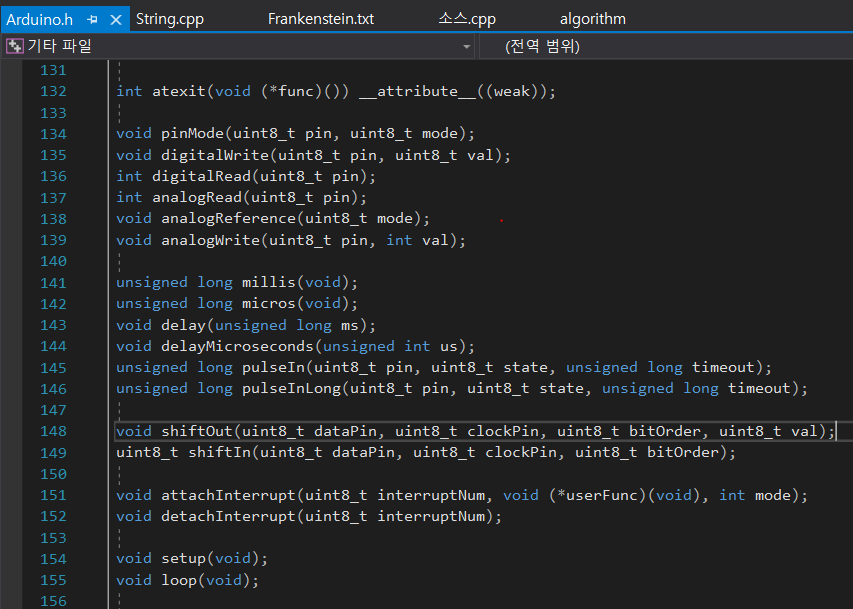
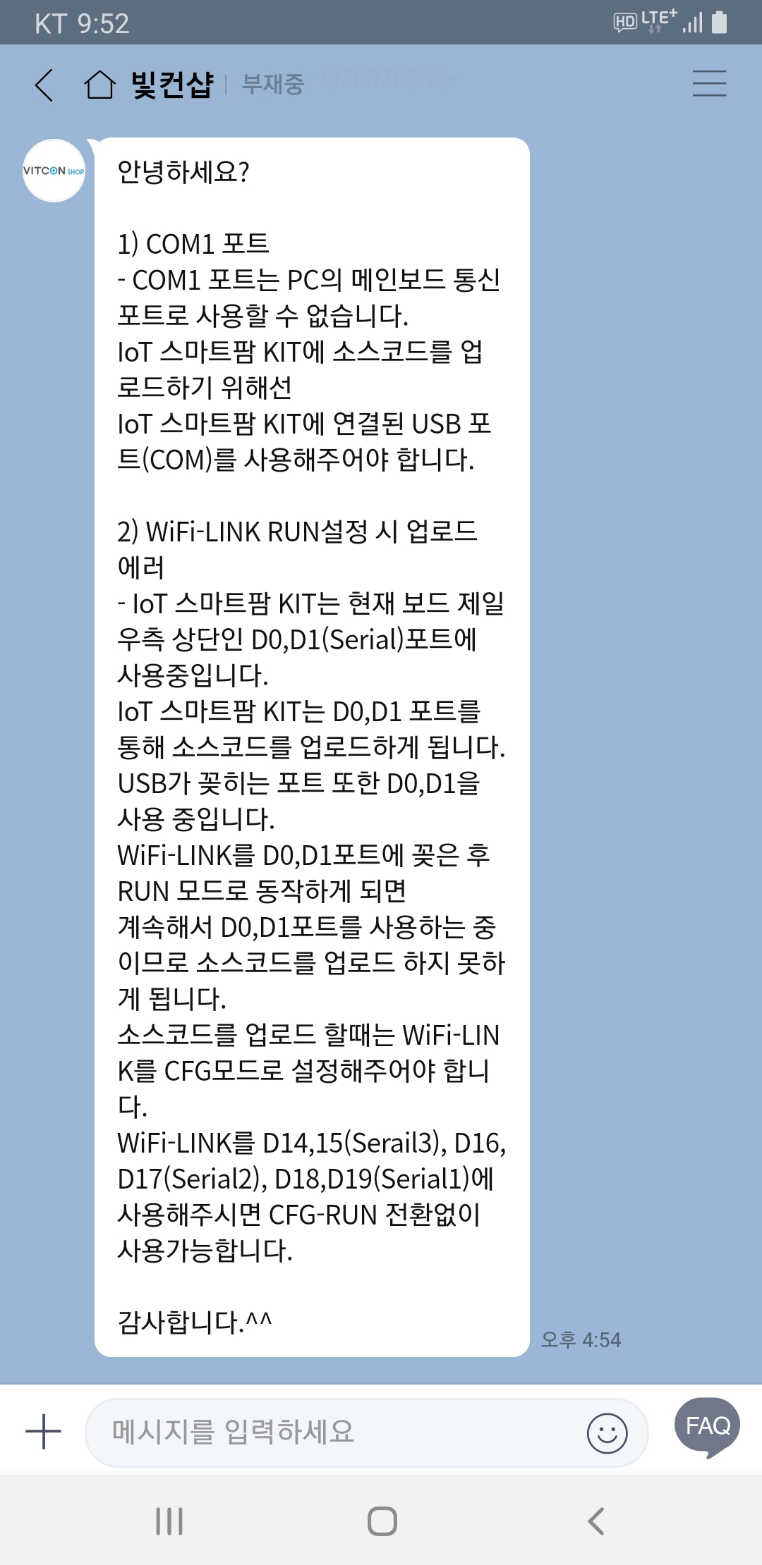


그림 1 Arduino.h

pinMode와 디지털 ,아날로그 신호 입출력 millis delay함수 등 기본적인 함수가 보이고 setup과 loop 또한 보임

=================================================================================



7. COM 왜 1은 하면 안되는지

+시리얼 연결 참고

COM1은 사용하지 말라는 의미는 시리얼포트인 COM1을 사용하지 말라는 의미.

COM1포트는 산업 장비와 옛날컴퓨터에서 볼 수 있는 포트 

스마트 팜을 업로드 하기 위해서는 시리얼 포트가 아닌, USB포트(COM)을 사용해야한다. 실제 실험으로 USB포트를 COM1으로 바꿔 실행하면 정상적으로 잘 되는 것을 알 수 있음.

8. Wifi-LINK 자체적으로 CFG가 아닐때 코드가 업로드 안되게 하는 기능이 있는지

답장 참고

=================================================================================

9. 내가 만든 아두이노 라이브러리 업로드 하는 방법

 GitHub에 Library Repository 등록

 GitHub Repository 폴더 구조는 아래과 같은 구조로 구성

* examples : 라이브러리 예제
* src : 라이브러리를 구성하는 전체 소스 파일 (.h, .cpp)
* README.md : 라이브러리를 설명하는 문서
* keywords.txt : 아두이노 IDE에서 Highlite 할 문자 표시
* library.properties : 라이브러리 이름, 버전, 저자, 및 다운로드 url

 라이브러리 Tag 생성  
  
출처: <https://kaizen8501.tistory.com/117> [Life4IoT]

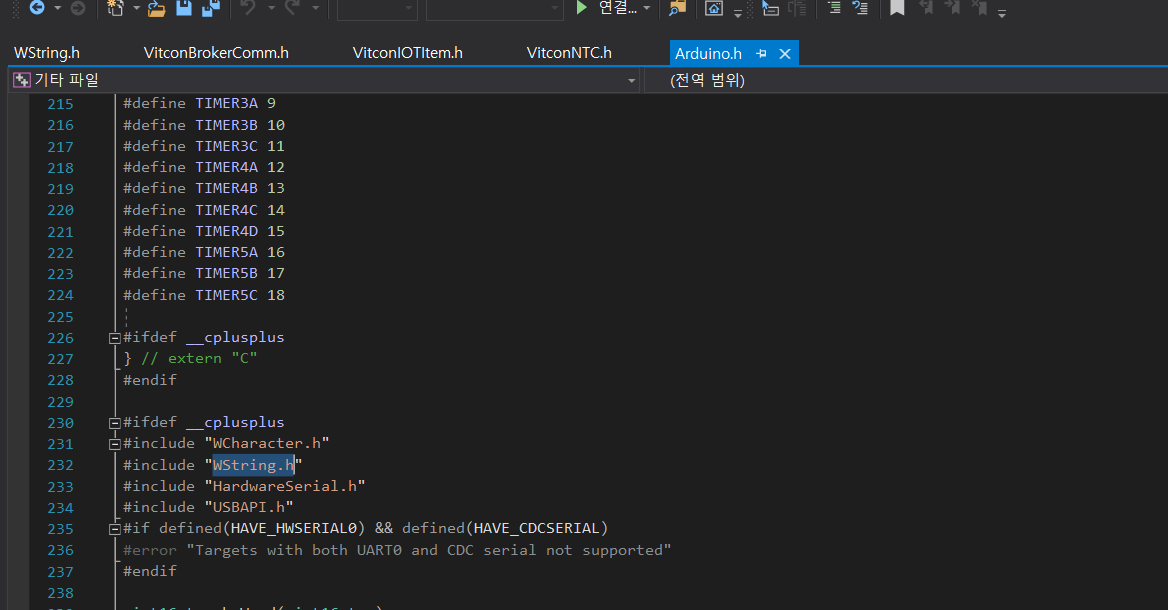
<https://m.blog.naver.com/PostView.nhn?blogId=junwha0511&logNo=221535738246&categoryNo=1&proxyReferer=https:%2F%2Fwww.google.com%2F>

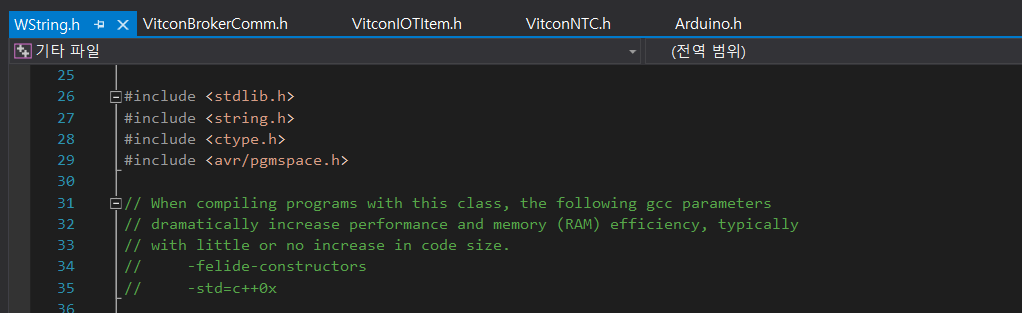
=================================================================================

10. 기본 클래스(String, Serial)들은 어디에 저장되어있는지

**String은 어디있나**

C:\Program Files\WindowsApps\ArduinoLLC.ArduinoIDE\_1.8.39.0\_x86\_\_mdqgnx93n4wtt\hardware\arduino\avr\cores\arduino 폴더안의

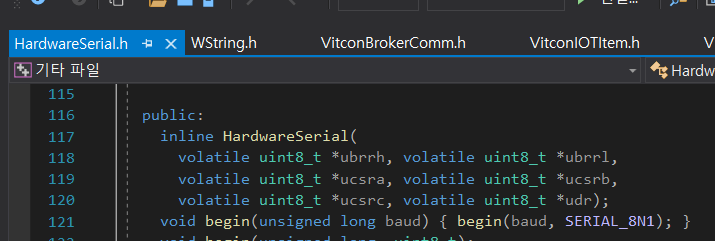




String이 있는 것을 볼 수 있다.

**Sereal은 어디있나**

C:\Program Files\WindowsApps\ArduinoLLC.ArduinoIDE\_1.8.39.0\_x86\_\_mdqgnx93n4wtt\hardware\arduino\avr\cores\arduino 폴더안의 아래 사진을 통해 있는 것을 볼 수 있다.



=================================================================================

11. 전처리기가 맨 뒤에 있어도 정상작동 되는가

안 돌아감.

C 컴파일 4단계

C는 컴파일된 프로그래밍 언어로서, 실행 시 대신 '컴파일 시간'에 기계에 의해 해석된다는 것을 의미한다.

SETP1 전처리기

전처리기의 일은 소스 코드를 가져다가 소스 코드와 로컬 매크로를 기반으로 출력을 생성하는 것이다.

그런 다음 출력은 컴파일러로 전송된다.

STEP2 컴파일러

코드를 기계가 이해할 수 있는 어셈블리 코드로 변경하는 것이다.

컴파일러의 출력은 다음 단계로 연결되는 조립 코드다.

SETP3 어셈블러

조립자는 조립코드를 조립하는 것으로서 조립코드를 1과 0으로 순차적으로 변환한다:

0101000 01001110. 조립자가 완성되면 출력이 링커로 전송된다

STEP 4 링커

링커는 당신의 머리글에 지정된 라이브러리와 당신의 소스 코드를 함께 연결하는 부분이다.

라이브러리에서 필요한 부품은 하나의 포장된 출력 패키지를 만들기 위해 컴파일된 코드와 결합된다.

#define은 메모리에 올라가지 않는다는 장점,

전처리기의 동작은 컴파일러 진입 이전에 끝나기 때문에 C언어의 문법과 키워드에 영향을 받지 않는다. 따라서  C언어의 do 와 같은 키워드도 전처리기에서 재정의 할 수 있다.

LINK = <https://medium.com/@holdengrissett/the-4-steps-of-c-compilation-72e8ec474bd5>

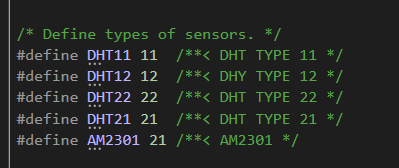
로컬매크로 참고LINK=<https://modoocode.com/99>

=================================================================================

12. DHT22 옆에 AM숫자 시리즈들은 뭔지

DHT22모듈에 사용되는 센서명이 AM2302.

DHT22 외에도 DHT11, DHT21등이 있으며 모듈마다 사용되는 센서 또한 다름

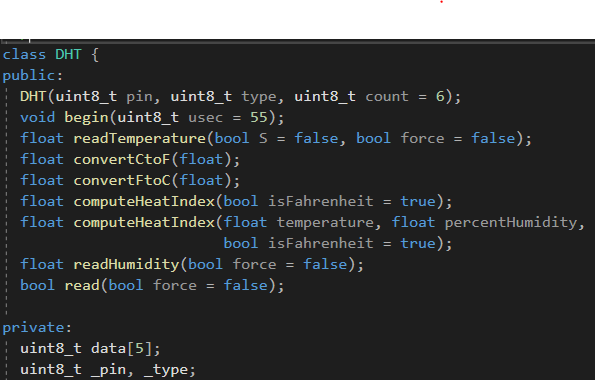
DHT11 DHT22 등 센서 타입의 정의를 볼 수 있다.

DHT 22 (AM2302)

실험 = #define AM2302 22를 하나 추가 하면

잘 돌아간다

13. DHT 클래스 안에 어떤 함수들이 있는지



DHT의 함수 목록 :

* 시작
* 온도 읽어 오기
* 섭씨를 화씨로 바꾸기
* 화씨를 섭씨로 바꾸기
* 습도 읽어 오기

14. dht() 안에 왜 굳이 디파인으로 썼는지

아두이노에서는 메모리의 용량이 PC와는 다르게 상당히 부족하여, const와 같은 변수를 많이 사용하게 되면 효율적이지 못하게 된다  
즉, 아두이노와 같은 임베디드 기기 특성상 메모리를 많이 사용하는 const보다는 #define 이 좋다.

15. Serial 함수에서 Serial의 의미

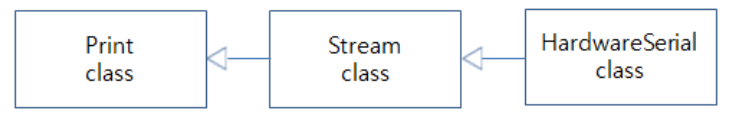
= 시리얼 통신을 의미, begin은 시리얼 통신을 시작한다는 뜻

1. Serial 객체 생성

Serial 객체는 어떤 것일까? HardwareSerial 클래스(class)를 미리 인스턴스로 만들어 놓은 객체가 Serial입니다. 스케치를 컴파일하면 스케치 파일 앞에 #include "Arduino.h"이 추가되고, Arduino.h 헤더(header) 파일에 #include "HardwareSerial.h"가 포함되고, HardwareSerial.h 파일에서 Serial 객체를 extern으로 선언하였고, HardwareSerial.cpp에서 Serial 객체를 만듭니다.



Serial은 stream클래스를 부모로 , Stream은 print를 부모로 아래 표와 같은 부모자식 관계에 있다.



**HardwareSerial의 멤버 함수**

**======================**

void begin(unsigned long);

void begin(unsigned long, uint8\_t);

void end();

virtual int available(void);

virtual int peek(void);

virtual int read(void);

virtual void flush(void);

virtual size\_t write(uint8\_t);

inline size\_t write(unsigned long n) { return write((uint8\_t)n); }

inline size\_t write(long n) { return write((uint8\_t)n); }

inline size\_t write(unsigned int n) { return write((uint8\_t)n); }

inline size\_t write(int n) { return write((uint8\_t)n); }

**Stream의 멤버 함수**

**======================**

void setTimeout(unsigned long timeout);

bool find(char \*target);

bool find(char \*target, size\_t length);

bool findUntil(char \*target, char \*terminator);

bool findUntil(char \*target, size\_t targetLen, char \*terminate, size\_t termLen);

long parseInt();

float parseFloat();

size\_t readBytes( char \*buffer, size\_t length);

size\_t readBytesUntil( char terminator, char \*buffer, size\_t length);

String readString();

String readStringUntil(char terminator);

**Print의 멤버 함수**

**======================**

size\_t print(const \_\_FlashStringHelper \*);

size\_t print(const String &);

size\_t print(const char[]);

size\_t print(char);

size\_t print(unsigned char, int = DEC);

size\_t print(int, int = DEC);

size\_t print(unsigned int, int = DEC);

size\_t print(long, int = DEC);

size\_t print(unsigned long, int = DEC);

size\_t print(double, int = 2);

size\_t print(const Printable&);

size\_t println(const \_\_FlashStringHelper \*);

size\_t println(const String &s);

size\_t println(const char[]);

size\_t println(char);

size\_t println(unsigned char, int = DEC);

size\_t println(int, int = DEC);

size\_t println(unsigned int, int = DEC);

size\_t println(long, int = DEC);

size\_t println(unsigned long, int = DEC);

size\_t println(double, int = 2);

size\_t println(const Printable&);

size\_t println(void);

Link=<https://m.blog.naver.com/PostView.nhn?blogId=msyang59&logNo=220047851579&proxyReferer=https:%2F%2Fwww.google.com%2F>

16. 시리얼 프린트에서 F의 의미

메모리 관리를 위해 문자열 등을 아두이노의 데이터 메모리(SRAM)에 저장하지 않고 프로그램 메모리(Flash)에 저장할 수 있도록 F() 함수를 이용

17. U8g2 초기화식 의미, u8g2 클래스 함수들

U8G2\_SH1106\_128X64\_NONAME\_F\_HW\_I2C u8g2(U8G2\_R0,/\*reset=\*/U8X8\_PIN\_NONE);

U8G2는 OLED 모듈 이름과 해상도에 따라 클래스 객체를 선언

현재 비트콘의 IoT-MODLINK-EX 보드에서 사용되는 OLED 모듈은

128\*64 SH1106모듈, HW I2C 핀을 사용

U8G2\_SH1106\_128X64\_NONAME\_F\_HW\_I2C 클래스 객체를 선언하게 됩니다.

클래스 인자로는 회전각도, 리셋핀입니다.

회전각도는 U8G2\_R0, U8G2\_R1, U8G2\_R2, U8G2\_R3, U8G2\_MIRROR

U8G2\_R0는 회전각도 없음, U8G2\_R1는 90도, U8G2\_R2은 180도, U8G2\_R3은 270도

U8G2\_MIRROR는 좌우반전입니다.

I2C 통신을 하므로 리셋은 선언해주지 않습니다.

clearBuffer () 메모리 프레임 버퍼의 모든 픽셀을 지웁니다.

drawBox () 상자를 채웁니다 (채워진 프레임).

drawCircle () 원을 그립니다.

drawDisc () 채워진 원을 그립니다.

drawEllipse () 타원을 그립니다.

drawFilledEllipse () 채워진 타원을 그립니다.

drawFrame () 프레임을 그립니다 (빈 상자).

drawGlyph () 단일 문자를 그립니다.

drawHLine () 수평선을 그립니다.

drawLine () 두 점 사이에 선을 그립니다.

drawPixel () 픽셀을 그립니다.

drawRBox () 둥근 모서리가 있는 상자를 그립니다.

drawRFrame () 둥근 모서리가 있는 프레임을 그립니다.

drawStr () 문자열을 그립니다.

drawTriangle () 삼각형 (채워진 다각형)을 그립니다.

drawUTF8 () UTF-8로 인코딩 된 문자열을 그립니다.

drawVLine () 세로선을 그립니다.

drawXBM () XBM 비트 맵을 그립니다.

getAscent () 기준선 이상의 글리프의 참조 높이를 반환합니다.

getDescent () 기준선 이하의 글리프의 참조 높이를 반환합니다.

getStrWidth () 문자열의 픽셀 너비를 반환합니다.

getUTF8Width () UTF-8로 인코딩 된 문자열의 픽셀 너비를 반환합니다.

sendBuffer () 메모리 프레임 버퍼의 내용을 디스플레이로 보냅니다.

setBitmapMode () 비트 맵 배경색 모드를 정의하십시오.

setContrast () 대비 또는 밝기를 설정하십시오.

setDisplayRotation () 디스플레이 회전을 변경합니다.

setDrawColor () 모든 그리기 기능에 대한 비트 값 (색상 인덱스)을 정의합니다.

setFlipMode () 플립 (180도 회전) 모드를 설정하십시오.

setFont () 글리프 및 문자열 그리기 기능을위한 u8g2 글꼴을 정의하십시오.

setFontDirection () 모든 문자열 또는 글리프의 그리기 방향을 설정하십시오.

setFontMode () 글꼴 배경색 모드를 정의하십시오.

setFontPosBaseline () 글리프 및 문자열 그리기 함수의 참조 위치를 "기준선"으로 변경하십시오.

setFontPosBottom () 글리프 및 문자열 그리기 기능의 참조 위치를 "bottom"으로 변경하십시오.

setFontPosTop () 글리프 및 문자열 그리기 기능의 참조 위치를 "상단"으로 변경하십시오.

setFontPosCenter () 글리프 및 문자열 그리기 기능의 참조 위치를 "중앙"으로 변경하십시오.

setFontRefHeightAll () 상승 및 하강 계산 모드를 "최고 및 최저 글리프"로 설정하십시오.

setFontRefHeightExtendedText () 상승 및 하강 계산 모드를 "최고 [A1 (], 최저 [g (])"로 설정하십시오.

setFontRefHeightText () 상승 및 하강 계산 모드를 "최고 [A1], 최저 [g]"로 설정하십시오

setPowerSave () 디스플레이의 절전 모드를 활성화 또는 비활성화합니다.

updateDisplay () 디스플레이를 업데이트합니다.

updateDisplayArea () 디스플레이의 지정된 사각형 영역을 업데이트합니다.