nipa 정보통신산업진흥원
National IT Industry Promotion Agency

페블 스마트와치 101

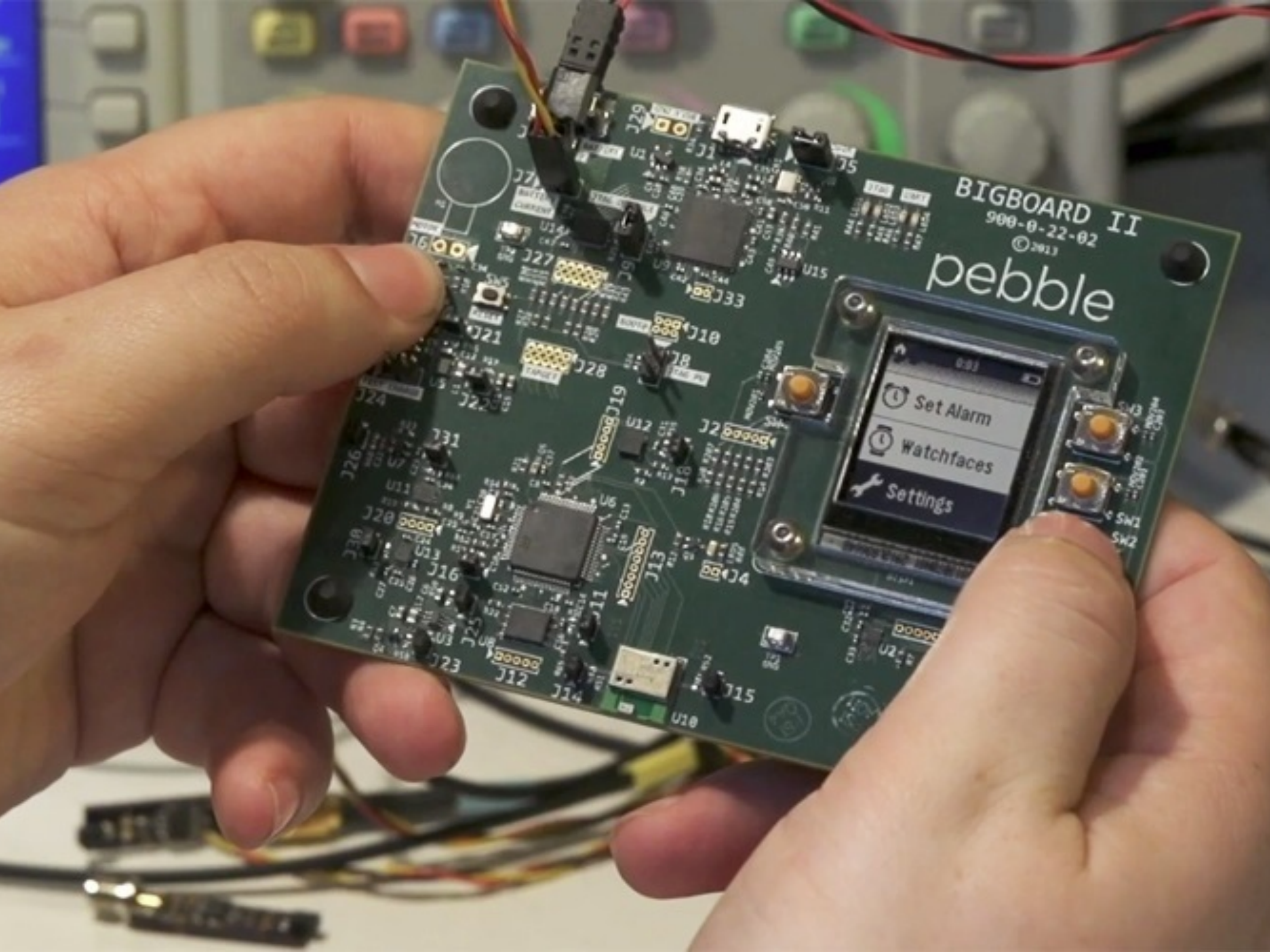
오픈프론티어 2기 주영택

공개SW개발자센터
OPEN FRONTIER LAB

- 휴대폰 보다 더 가까운 차세대 스마트기기로 관심이 주목되는 스마트와치 부분의 작지만 무시할 수 없는 페블의 구조와 종류별 특징을 살펴보고 응용 프로그램 환경과 개발 환경을 통해 페블 생태계를 살펴봅니다.

- 간단한 와치페이스 소스코드와 함께 기본적인 코드 구조를 확인하고 페블 애플리케이션의 특징을 설명합니다. 에뮬레이터와 실 기기에 빌드하고 페블 앱스토어에 배포하는 과정을 소개합니다.

- 한국어 언어팩에 대한 이야기와 함께 국내 환경을 기준으로 다른 스마트와치 플랫폼과 페블의 장단점 대해 비교해 봅니다.


BIGBOARD II
900-0-22-02
pebble
Set Alarm
Watchfaces
Settings


Wednesday
12:11
April 2
13°
Professorville 12:09
pebble
9:45
Moshpit
We carry the
heart (Live...
2:34
3:20
5:42AM
Sunrise
75°/65°
7:30AM
Coffee Date

Units in Millions

Operating System	2015 Unit Shipments	2015 Market Share	2019 Unit Shipments	2019 Market Share	2015-2019 CAGR
Android	0.4	2.1%	1.1	1.2%	25.8%
Android Wear	3.2	15.2%	34.3	38.8%	80.5%
Linux	0.2	0.9%	1.1	1.2%	54.5%
Pebble OS	**1.8**	**8.6%**	**2.3**	**2.6%**	**5.8%**
RTOS	0.8	3.8%	1.9	2.2%	23.8%
Tizen	1.7	8.2%	2.5	2.8%	9.5%
WatchOS	13.0	61.3%	45.2	51.1%	36.5%
Total	**21.3**	**100.0%**	**88.3**	**100.0%**	**42.8%**

http://www.idc.com/getdoc.jsp?containerId=prUS40846515

Pebble Time Round 14mm

Silver with Sto

$249.99 USD

Pebble Time Round 20mm

Black with Ne

$249.99 USD

Pebble Time Steel

Silver with Lea

$249.99 USD

Pebble Time

Red

$149.99 USD

Pebble Steel

Matte Black w

$149.99 USD

Pebble Classic

Jet Black

$99.99 USD

구분	페블 클래식	페블 스틸	페블 타임	페블 타임 스틸	페블 타임 라운드
출시	2013년 말	2014년 초	2015년 전반기	2015년 중순	2015년 하반기
무게(g)	38	56	42	62 / 116	28 / 32
밴드(mm)	22	22 (custom)	22		14 / 20
크기(mm)	52 x 36 x 11.5	46 x 34 x 10.5	40.5 x 37.5 x 9.5	40.5 x 37.5 x 9.5	38.5 x 38.5 x 7.5
화면	1.26" 흑백		1.25" 컬러		원형 1.25" 컬러
해상도(px)	144 x 168				180 x 180 (+7% area)
피드백	전동 모터·스피커 없음				
센서	지자계, 주변광, 가속도(3축)				
통신	블루투스 2.1 + 4.0 (EDR,LE)				
마이크	없음		내장 마이크		
방수	5기압·40미터				생활방수(샤워불가)
프로세서(Mhz)	Cortex-M3 (64)		ARM Cortex-M4 (100)		
메모리(KB)	512		1024		
스토리지(MB)	8		16 (+1 for System)		
배터리(mAh)	130		150	200 (approximately)	·
확장	없음		스마트스트랩		

구분	페블 클래식	페블 스틸	페블 타임	페블 타임 스틸	페블 타임 라운드
플랫폼	Aplite		Basalt		Chalk
코드(KB)	24		64		
리소스(KB)	96		256		
컬러	2		64		
버튼	4				
SDK	2.x (올해 말 업그레이드)		3.x		

3.x apps are compiled per platform

compiled against		runs on: 2.x Aplite	runs on: 3.x Aplite	runs on: 3.x Basalt	runs on: 3.x Chalk
2.x	Aplite	✅	✅	✅	❌
3.x	Aplite	❌	✅	❌	❌
3.x	Basalt	❌	❌	✅	❌
3.x	Chalk	❌	❌	❌	✅

PEBBLE OPEN SOURCE

Here at Pebble we are big believers in the power of open source. Our firmware is built upon FreeRTOS, and all of our mobile and web applications make use of many different open source libraries.

Whenever possible, we like to give back to the open source community, so here are just some of our open source projects available for anyone to use in their own work.

You can also check out our GitHub profile at https://github.com/pebble/.

CloudPebble

CloudPebble
Web-based IDE for Pebble development.

CloudPebble QEMU Controller
Controls QEMUs

CloudPebble ycmd Proxy
Handles communication with the autocompletion daemon(s) on behalf of CloudPebble clients.

Pebble SDK

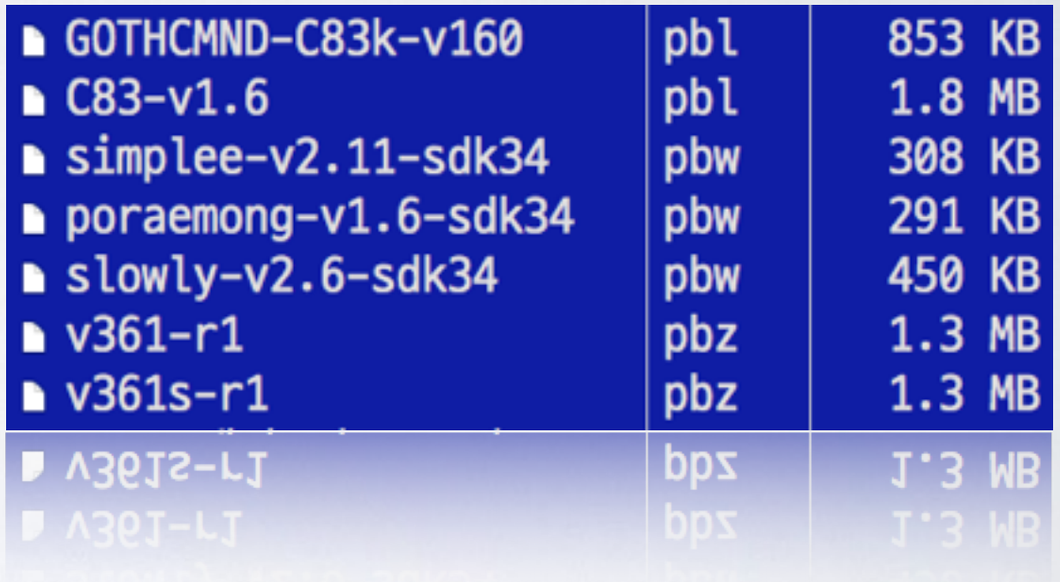
libpebble2
Library for communication with Pebble devices and emulators

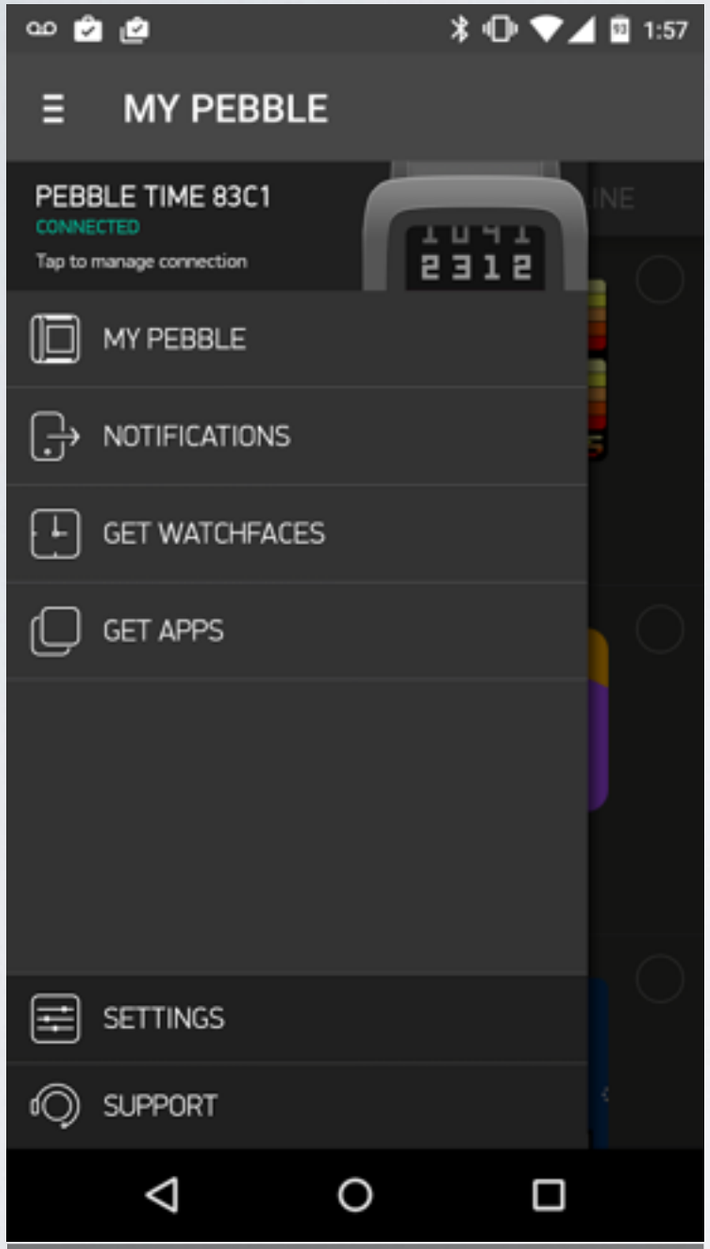
pebble-tool
The Pebble tool, as distributed in the Pebble SDK

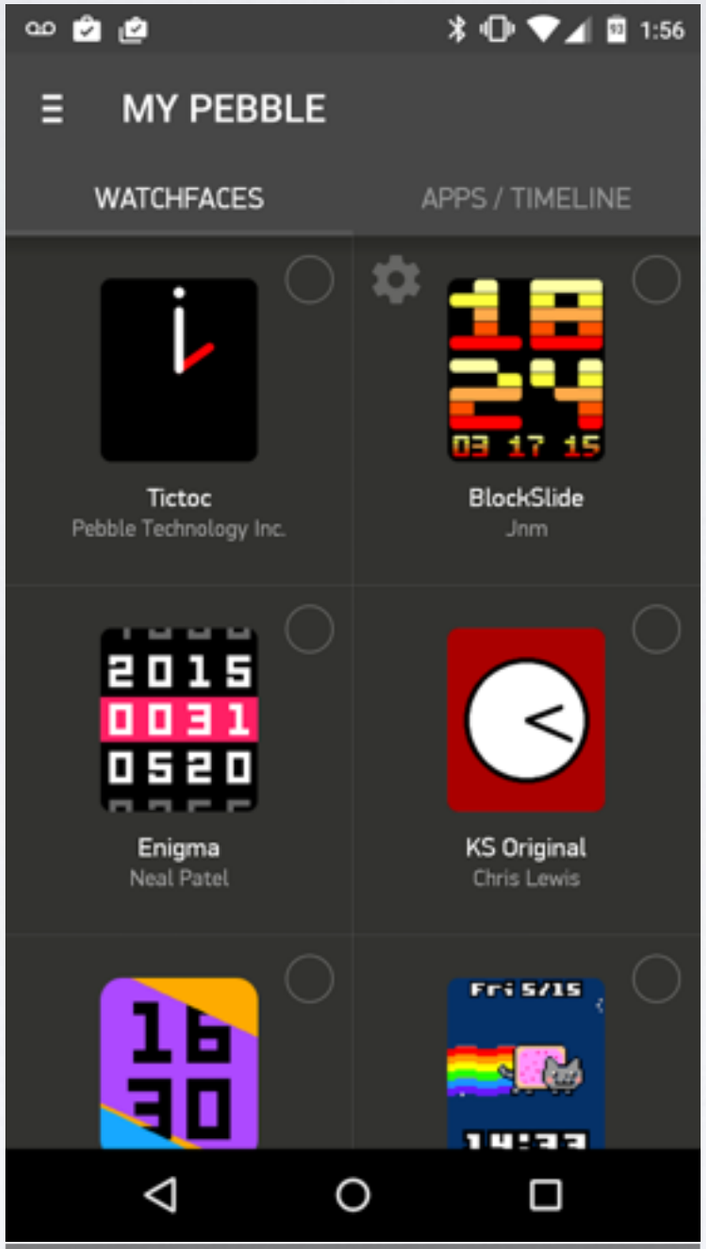
WORK FOR US!

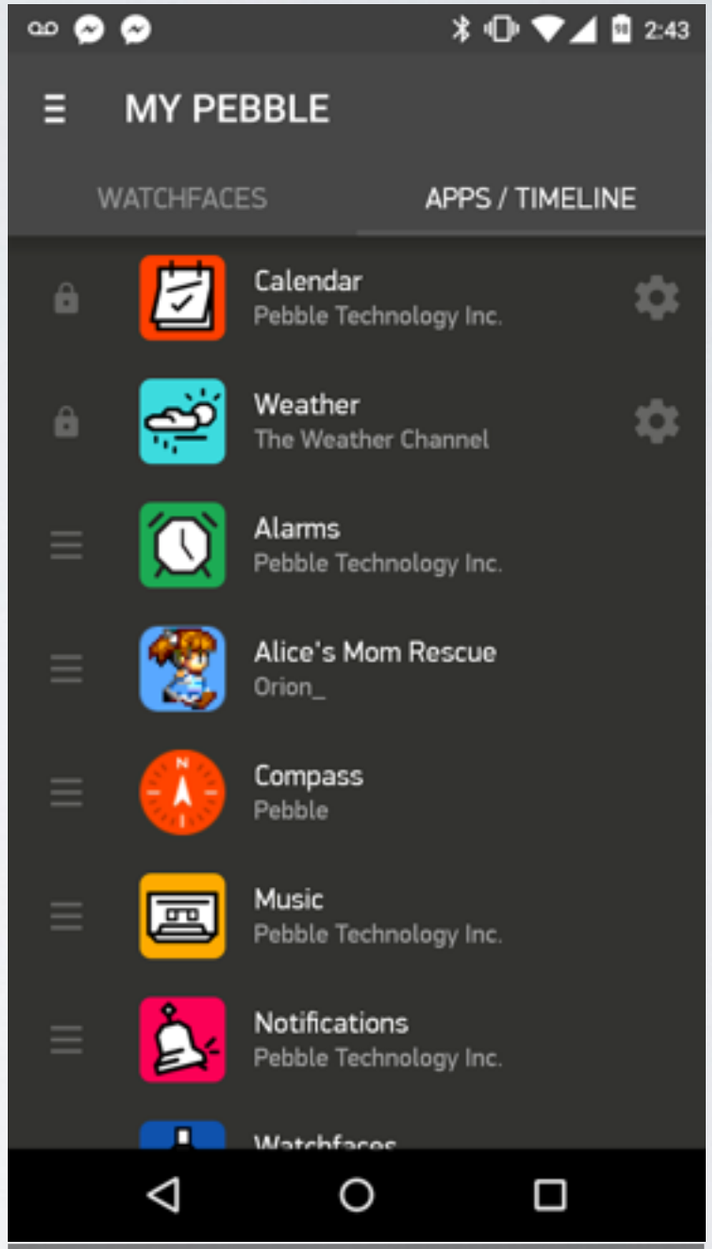
As a rapidly growing company, we're always on the lookout for new talent to grow our various teams. If you or anyone you know has the technical skills, wearable enthusiasm and a taste for the Valley lifestyle, have a look at our jobs page to see a list of open positions.

- PBZ : 펌웨어
- PBL : 언어팩
- PBW : 애플리케이션 파일




1:57
MY PEBBLE
PEBBLE TIME 83C1
CONNECTED
Tap to manage connection
MY PEBBLE
NOTIFICATIONS
GET WATCHFACES
GET APPS
SETTINGS
SUPPORT


1:56
MY PEBBLE
WATCHFACES
APPS / TIMELINE
Tictoc
Pebble Technology Inc.
BlockSlide
Jnm
Enigma
Neal Patel
KS Original
Chris Lewis


2:43
MY PEBBLE
WATCHFACES
APPS / TIMELINE
Calendar
Pebble Technology Inc.
Weather
The Weather Channel
Alarms
Pebble Technology Inc.
Alice's Mom Rescue
Orion_
Compass
Pebble
Music
Pebble Technology Inc.
Notifications
Pebble Technology Inc.

Pebble timeline

PAST

PRESENT

FUTURE

Materials

@Neal Patel

@Chris Lewis

Slider

@chadheim

Weather Land

@reno

Squared 3.0

@hexahedria

Smile

@dezign999

Watch apps you'll love.

ESPN

UBER

runkeeper

The Weather Channel

PANDORA

Swarm

MISFIT

EVERNOTE

UP by JAWBONE

GoPro

iHeart RADIO

Domino's








WRITING APPS FOR PEBBLE

This section of the Developer Guides contains lots of information on writing watchfaces for the Pebble platform. New and existing developers will find de aspects of development from basic app structure to listening for system eve more advanced topics like communication and using the built-in sensors.

In addition to these guides, there are also several tutorials available to help started writing their own watchapps.

Contents

These sections are outlined below:

- **Pebble App Basics** - The basic structure of all Pebble watchapps and wat
- **Adding Resources** - How to load and use custom resources in Pebble ap fonts, images, and raw files.
- **Graphics & Animations** - Details of all the APIs available for creating wat as using fonts, images, and animations.
- **Listening to System Events** - Details on all the available system event se be used to adapt an app's UI to user interaction and system events.
- **Communicating with the Phone** - How to use the various communicatio connect your watchapp with a companion phone app and web services.
- **Using Sensor Data** - How to use Pebble's onboard sensors, such as the a accelerometer.

```
Pebble Tool

Command:
    build               Builds the current project.
    clean
    new-project         Creates a new pebble project with the given name in a new directory.
    install             Installs the given app on the watch.
    logs                Displays running logs from the watch.
    screenshot          Takes a screenshot from the watch.
    insert-pin          Inserts a pin into the timeline.
    delete-pin          Deletes a pin from the timeline.
    emu-accel           Emulates accelerometer events.
    emu-app-config      Shows the app configuration page, if one exists.
    emu-battery         Sets the emulated battery level and charging state.
    emu-bt-connection   Sets the emulated Bluetooth connectivity state.
    emu-compass         Sets the emulated compass heading and calibration state.
    emu-control         Control emulator interactively
    emu-tap             Emulates a tap.
    emu-time-format     Sets the emulated time format (12h or 24h).
    ping                Pings the watch.
    login               Logs you in to your Pebble account. Required to use the timeline and Cloud
    logout              Logs you out of your Pebble account.
    repl                Launches a python prompt with a 'pebble' object already connected.
    transcribe          Starts a voice server listening for voice transcription requests from the
    data-logging        Get info on or download data logging data
    analyze-size        Analyze the size of your pebble app.
    convert-project     Structurally converts an SDK 2 project to an SDK 3 project. Code changes r
    kill                Kills running emulators, if any.
    wipe                Wipes data for running emulators. By default, only clears data for the cur
```

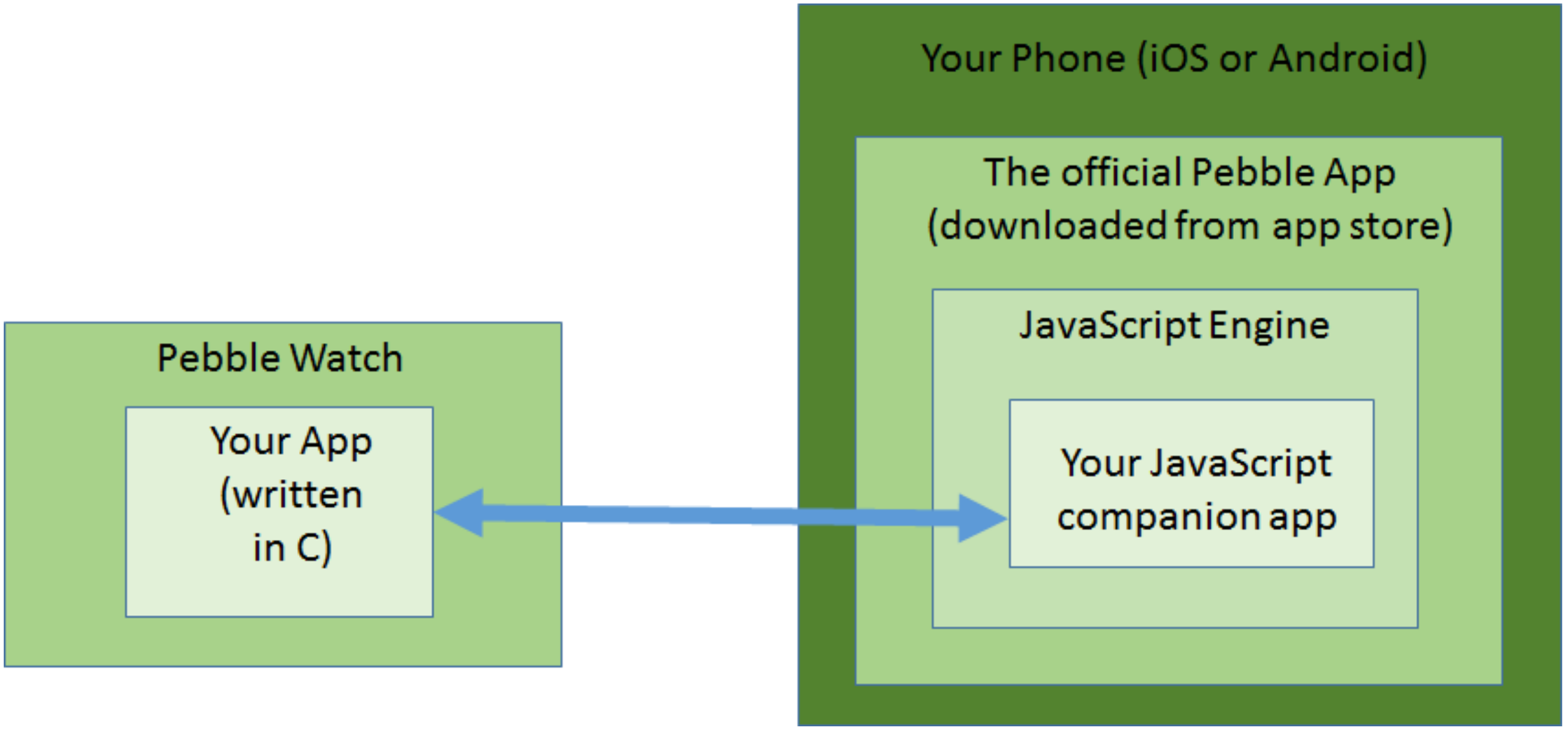
C99/11는 C 언어의 현대 개정판이다.

C99는 다음과 같은 기능들을 포함하고 있다. 이들 중 일부는 이미 일부 컴파일러에 확장 기능으로서 포함된 적이 있다.

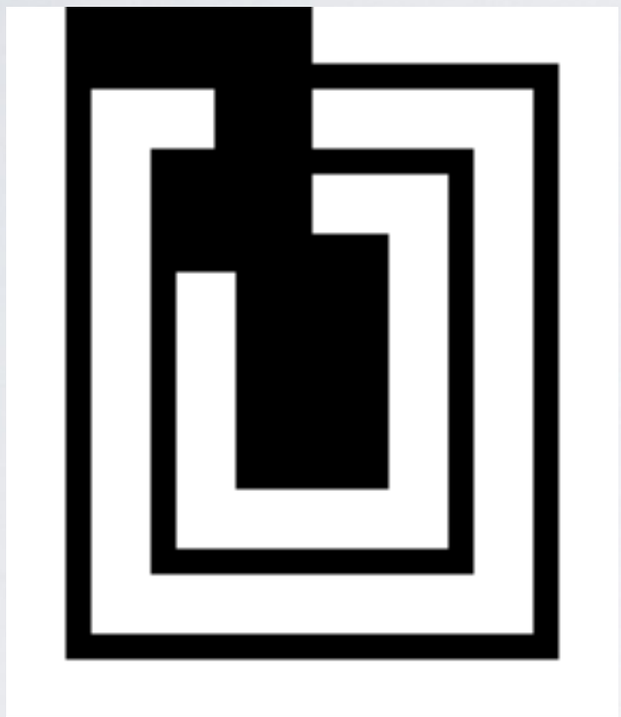
- 인라인 함수의 도입
- 변수의 선언은 더이상 파일 범위나 복합 명령어의 시작에서만 할 필요가 없다.
- long long int, 선택적인 확장 정수형, 명시적 불린 자료형, 그리고 복소수를 나타내기 위한 complex 자료형 등 새로운 자료형 도입
- 가변 길이 배열(VLA: variable-length array)
- BCPL이나 C++와 같은 //로 시작하는 주석들
- snprintf와 같은 새로운 라이브러리 함수
- stdbool.h 및 inttypes.h와 같은 새로운 헤더 파일들
- 자료형에 무관하게 동작하는(type-generic) 수학 함수들 (tgmath.h에 포함)
- IEEE 부동소수점 자료에 대한 개선된 지원
- 지정된 이니셜라이저(designated initializers)
- 복합 리터럴(compound literals)
- 가변 인수 매크로(Variadic macro)의 도입
- 보다 적극적인 코드 최적화를 위한 restrict 한정자

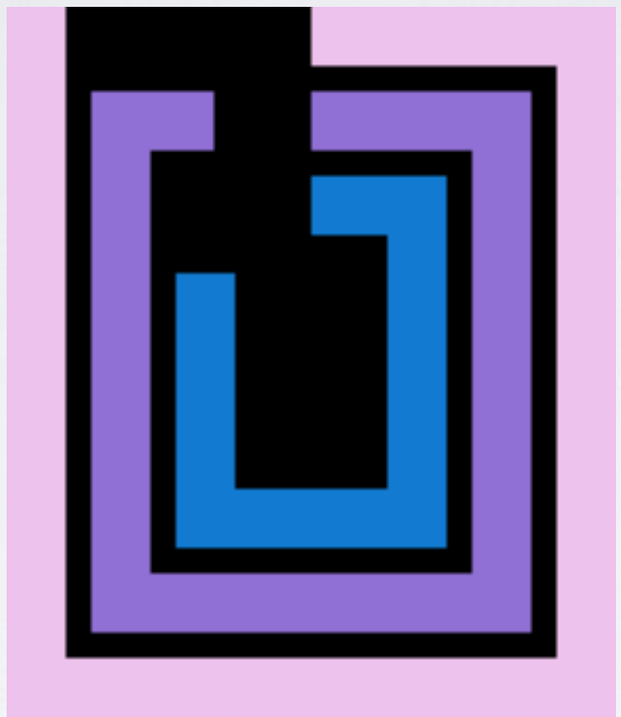
C99에서 바뀐 점

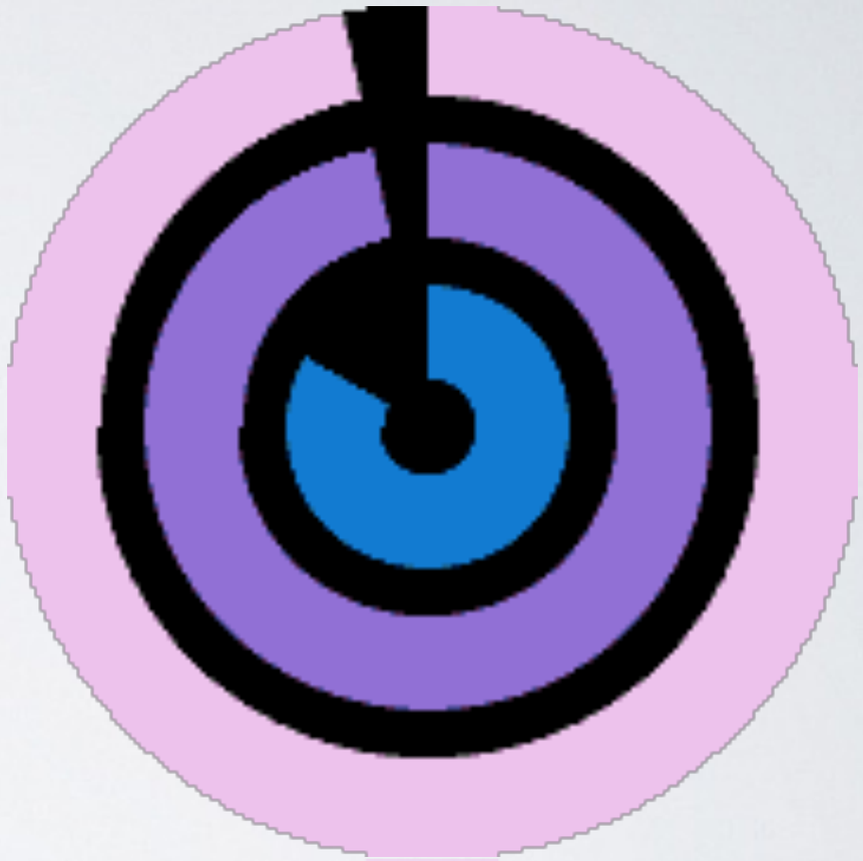
- 정렬 기능. (_Alignas 한정자, alignof 연산자, aligned_alloc 함수 지원 헤더 파일)
- _Noreturn 함수 한정자.
- 형에 따른 제네릭 기능으로 _Generic 키워드 추가. 예를 들면 다음 매크로 cbrt(x)는 x의 형에 따라 cbrtl, cbrtf, cbrt로 호출 됨.
- 멀티스레드 지원. (_Thread_local 스토리지 한정자, 헤더에 스레드 생성 관리 함수, 뮤텍스, 컨디션, 로컬 스레드 저장소 함수 지원. 또한 _Atomic - 형 한정자와 헤더를 이용해 아토믹 오퍼레이션 지원) C 유니코드 기술 문서 ISO/IEC TR 19769:2004를 토대로 유니코드 지원 개선 (char16_t와 char32_t를 각각 UTF-16/UTF-32 코드 저장용 형으로 할당. 유니코드 - 변환 함수를 에 추가 함)
- 메모리 경계 검사(Memory bounds checing) 인터페이스(Annex K).
- 이전 C99에서 구식 함수로 규정한 gets()를 제거. 대신 버퍼 안정화/메모리 경계 검사 기능을 추가한 gets_s를 표준에 편입함.
- 분석 기능의 추가 (Annex L).
- 부동 소수점(float) 형의 특징, 예를 들면 진수 변환, 정수부 얻기 등등을 검사하기 위한 매크로 추가
- 이름없는 union 또는 struct를 사용하기 더 편리해짐 예시: struct T { int tag; union { float x; int n; }; };.
- 정적 어셔션(Static assertion)은 이제 전처리기 #if, #error 등이 평가되기 전에 컴파일러가 먼저 형식을 처리한 다음 어셔션 처리가 수행됨
- fopen() 함수에 배타적 생성 모드 ("…x")가 생김. 이 기능은 POSIX open() 함수의 O_CREAT|O_EXCL기능과 같으며 보통 파일을 배타적으로 생성 할 때 유용하다.
- exit() 함수와 별개로 최소한 종료 작업만 수행하고 프로그램을 종료하는 새로운 함수 quick_exit()의 추가.
- 복소수를 생성하는 매크로의 추가 [5]


Pebble Watch
Your App
(written
in C)
Your Phone (iOS or Android)
The official Pebble App
(downloaded from app store)
JavaScript Engine
Your JavaScript
companion app

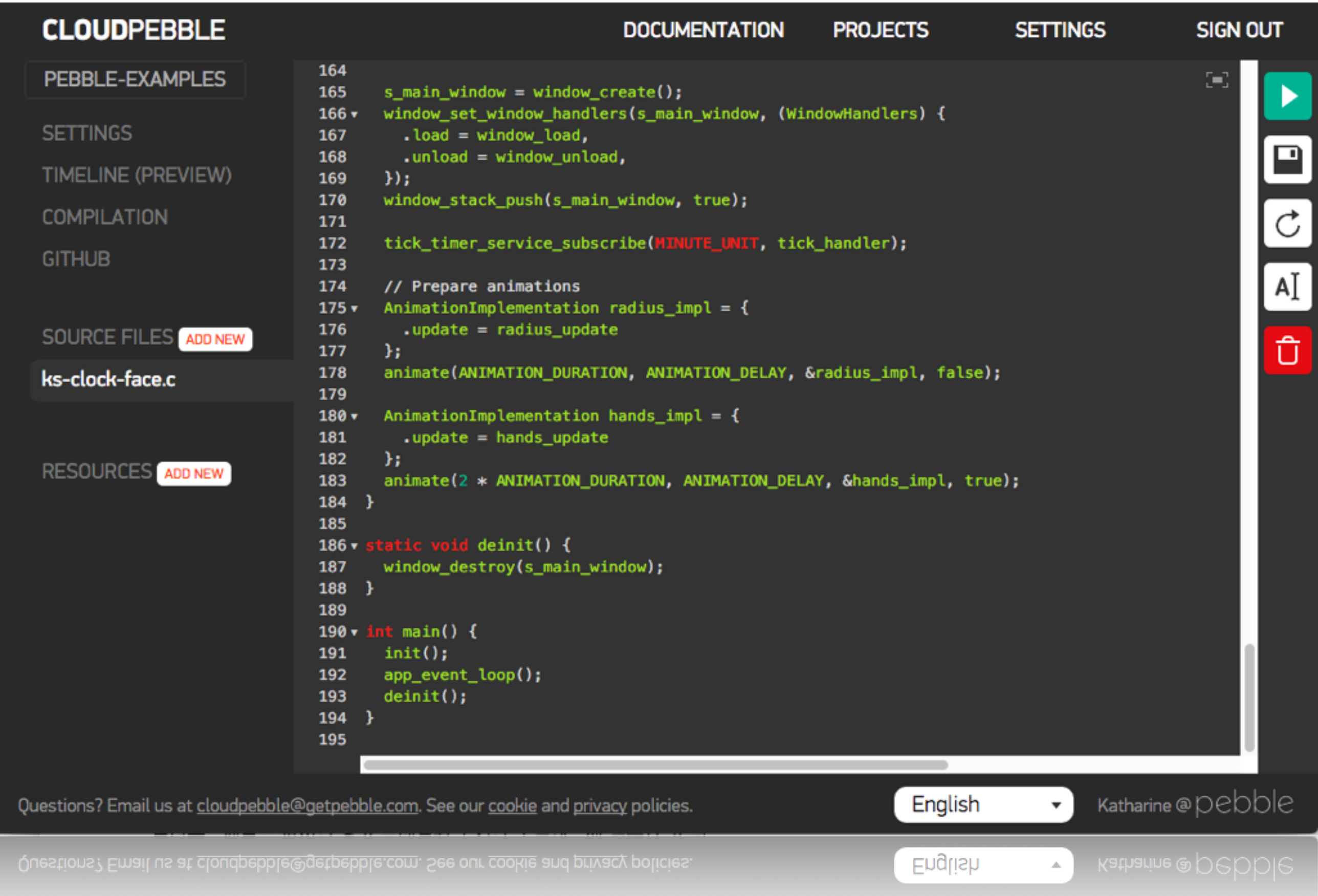
- APLITE: 페블 클래식 (스틸 포함)
  - 흑백 화면, 사각형 화면
- BASALT: 페블 타임 (타임 스틸 포함)
  - 64컬러 화면, 사각형 화면, 마이크 입력, 스트랩 확장
- CHALK: 페블 타임 라운드
  - 64컬러 화면, 원형 화면, 마이크 입력, 스트랩 확장







http://cloudpebble.net



https://www.youtube.com/watch?v=zwH4T8OaXyI

```c
#include <pebble.h>

static void init() {
// 윈도우 및 레이어 등록·할당, 리소스 로딩
// 타이머 등록, 타이머 콜백 등록 등의 과정
}

static void deinit() {
// 리소스 해제, 윈도우 및 레이어 해제
}

int main(void) {
  init();
  app_event_loop();
  deinit();
}
```

```
static void init() {
  s_main_window = window_create();

  window_set_window_handlers(s_main_window, (WindowHandlers) {
    .load = main_window_load,
    .unload = main_window_unload
  });

  window_stack_push(s_main_window, true);

  tick_timer_service_subscribe(MINUTE_UNIT, tick_handler);
}
```

```
$ tree -L 1
├── appinfo.json
├── build/
├── readme.md
├── resources/
├── src/
└── wscript
```

```
APP_LOG(APP_LOG_LEVEL_INFO, "Main window loaded.");
```

```
[17:59:34] ocess_manager.c:369> Heap Usage for App <HannaClock:
           Total Size <63052B> Used <13852B> Still allocated <0B>
```

```
console.log(메시지);
Pebble.sendAppMessage(메시지 객체, 성공 콜백, 실패 콜백);
```

```
$ pebble logs --phone=폰 아이피
$ pebble logs --emulator 플랫폼명
```

```
tick_timer_service_subscribe(SECOND_UNIT, 핸들러)
```

ActionBarLayer

Vertical, bar-shaped control widget on the right edge of the window.

BitmapLayer

Layer that displays a bitmap image.

InverterLayer

Layer that inverts anything "below it".

MenuLayer

Layer that displays a standard list menu. Data is provided using callbacks.

RotBitmapLayer

Layer that displays a rotated bitmap image.

ScrollLayer

Layer that scrolls its contents, animated.

SimpleMenuLayer

Wrapper around MenuLayer, that uses static data to display a list menu.

TextLayer

Layer that displays and formats a text string.

StatusBarLayer

Layer that serves as a configurable status bar.

App entry point and event loop.

App is a module that provides you with an event loop for your Pebble app. All interaction between Pebble apps and the underlying Pebble OS takes place through an event loop. Before calling the app_event_loop() function, you subscribe to event services and implement event handlers. Each handler receives specific types of Events dispatched throughout the life of the Pebble watchapp.

The app_event_loop() function takes care of both waiting for new events to become available on the watchapp event bus and routing new events to the appropriate handler. Event Service allows an app to directly register for different types of events. This function will block until the watchapp is ready to exit, and should be placed in the app's main() function.

A watchapp typically configures and uses the app_event_loop() as follows:

```
int main(void) {
  // do set up here

  // Enter the main event loop. This will block until the app is ready to exit.
  app_event_loop();

  // do clean up here
}
```

DRAWING TEXT

Functions to draw text into a graphics context.

See Graphics Context for more information about the graphics context.

Other drawing functions and related documentation:

Drawing Primitives

Drawing Paths

Graphics Types

Function Documentation

SDK 2 | SDK 3

void **graphics_draw_text**(GContext * ctx, const char * text, GFont const font, const GRect box, const GTextOverflowMode overflow_mode, const GTextAlignment alignment, const GTextLayoutCacheRef layout)

Draw text into the current graphics context, using the context's current text color. The text will be drawn inside a box with the specified dimensions and configuration, with clipping occuring automatically.

PARAMETERS

ctx

▲ ▼ 12 : 30 ▲ ▼

열 한 다 세 네
두 여 섯 일 곱
여 덟 아 홉 시
정 각 이 삼 십
사 오 십 오 분

Anti-alias

Anti-alias

```
65,7,11,1,7,9
. . . O . . .
. . . O . . .
. . O . O . .
. . O . O . .
. . O . O . .
. O . . . O .
. O . . . O .
. O O O O O .
O . . . . . O
O . . . . . O
O . . . . . O
```

COLOR PICKER

Click a hexagon in the color map to see its SDK constant and how to use it in your Pebble app.

UNCORRECTED SUNLIGHT

Name:

Sample:

HTML code:

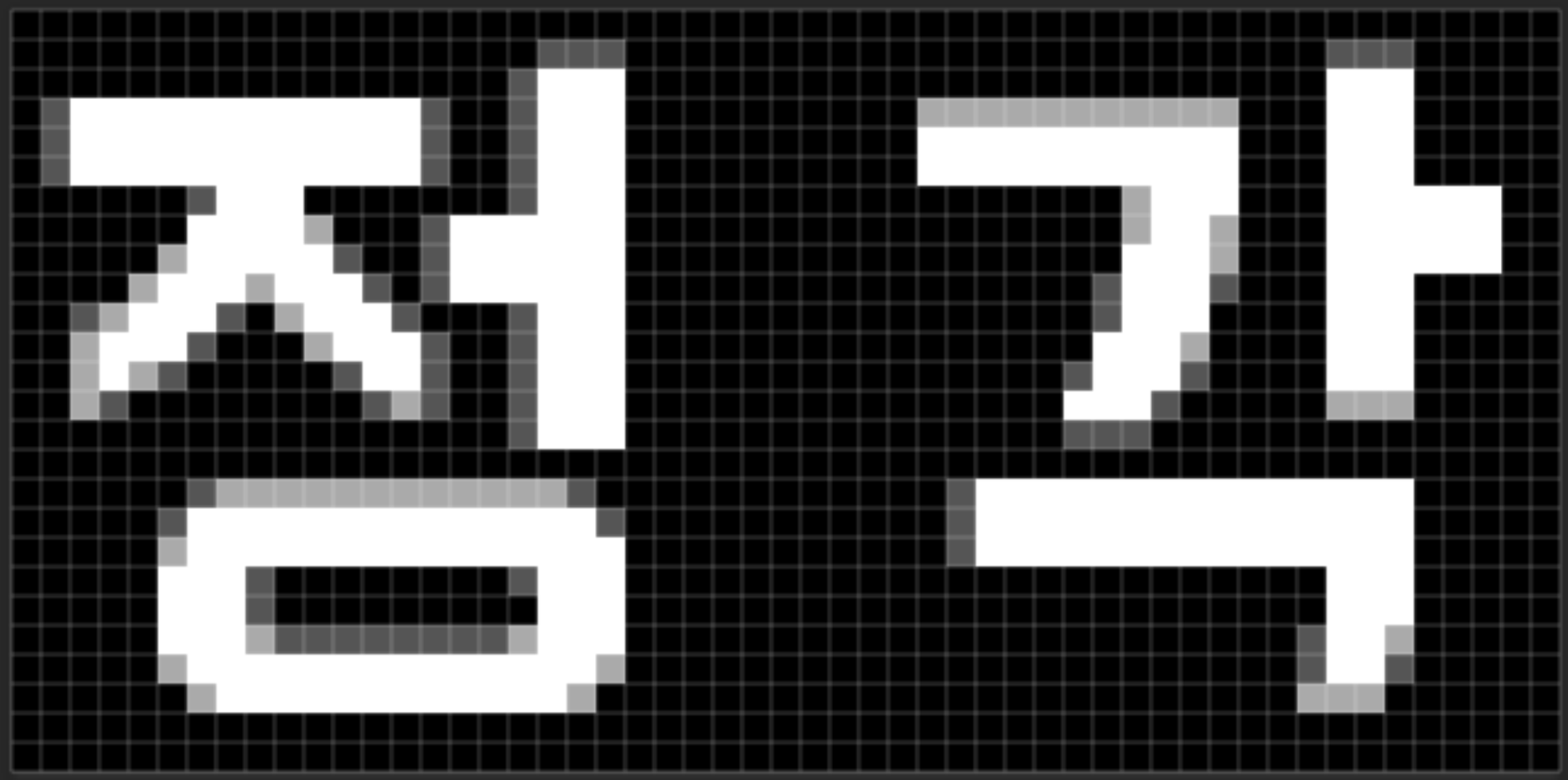
Uncorrected HTML code:

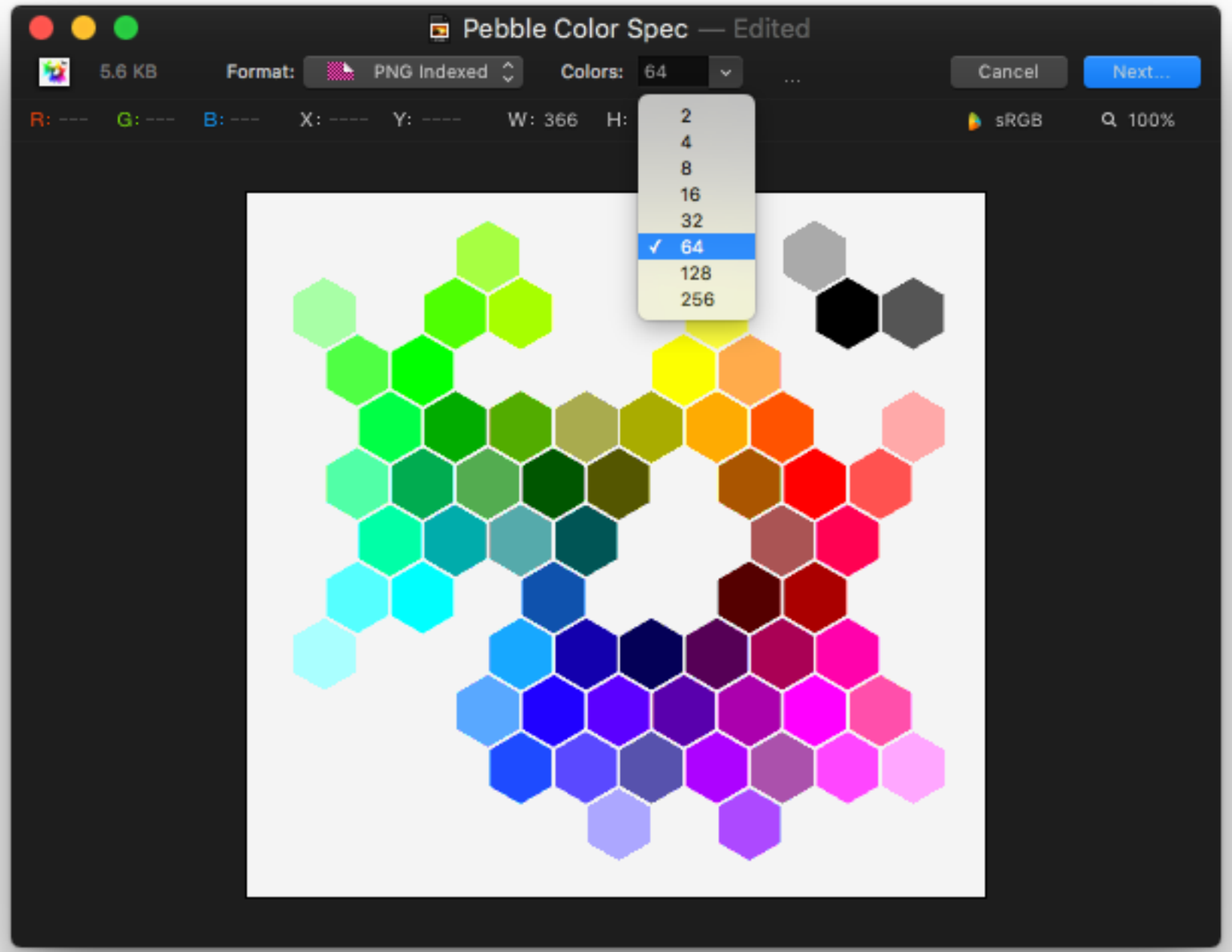
SDK Constant:

Code (RGB):

Code (Hex):

Example Code Segment




Pebble Color Spec — Edited
5.6 KB
Format:
PNG Indexed
Colors:
64
Cancel
Next…
R: ---
G: ---
B: ---
X: ----
Y: ----
W: 366
H:
sRGB
100%
2
4
8
16
32
✓ 64
128
256

Save for Web (100%)

Tip: Use File > Export > Export As... or right click on a layer for a faster way to export assets

Learn More

Original | Optimized | 2-Up | 4-Up

정각

Original: "o_clock.png"
4.04K

정각

PNG-8
294 bytes
1 sec @ 1 Mbps

0% dither
Custom palette
4 colors

정각

PNG-8
292 bytes
1 sec @ 1 Mbps

0% dither
Selective palette
4 colors

정각

PNG-8
292 bytes
1 sec @ 1 Mbps

88% dither
Selective palette
4 colors

100% R: -- G: -- B: -- Alpha: -- Hex: -- Index: --

Preview...

Preset: [Unnamed]

PNG-8

Custom Colors: Auto

No Dither Dither:

Transparency Matte: None

Amount:

Interlaced Web Snap: 0%

Embed Color Profile

Convert to sRGB

Preview: Monitor Color

Metadata: Copyright and Contact Info

Color Table

4

Image Size

W: 53 px

H: 26 px

Percent: 100 %

Quality: Bicubic

Animation

Looping Options: Once

1 of 1

Save... Cancel Done

빌드 & 인스톨 & 퍼블리쉬

컴파일 하기

```
pebble clean && pebble build
```

에뮬레이터에 인스톨 하기

```
pebble build && pebble install --emulator basalt
```

실 기기에 설치하기

```
pebble build && pebble install --phone 192.168.219.102
```

앱스토어에 퍼블리쉬하기, https://dev-portal.getpebble.com/

pebble

Pebble SDK Get Support Login Register

Publish your App in the Pebble appstore

The Pebble appstore is embedded in the official Pebble smartphone app. The easiest and best way to distribute your Pebble app to users.

The Pebble appstore is your portal for publishing and distributing your Pebble watchfaces, watchapps and Pebble companion apps worldwide.

Publishing in the Pebble appstore is free for all Pebble developers.

Pebble Appstore Guide Publish a Pebble App

HANNACLOCK

Soomtong

+ iOS | 27 | WATCHFACE

DESCRIPTION

Simple Watchface using Hanna font for Pebble Time

- Update Screen Every 5 min

Inspirited by [Toby Yun](http://tobyyun.com/wooclock/)

Hanna Font Created & Published by Woowahan.com

DEVELOPER	Soomtong
CATEGORY	Faces
UPDATED	Jul 25, 2015
VERSION	1.4

main.c UNREGISTERED

main.c

```c
#include <pebble.h>

// pair bitmaps with layers for just convenience
enum PNGBitmaps {
  bitmap_plate = 0,
  bitmap_hour_mark,
  bitmap_h1,
  bitmap_h2,
  bitmap_h3,
  bitmap_h4,
  bitmap_h5,
  bitmap_h6,
  bitmap_h7,
  bitmap_h8,
  bitmap_h9,
  bitmap_h10,
  bitmap_min_mark,
  bitmap_o_clock,
  bitmap_m5,
  bitmap_m10,
  bitmap_m20,
  bitmap_m30,
  bitmap_m40,
  bitmap_m50,
  bitmap_m10_2,
  bitmaps_length
};

static Window *window;
static BitmapLayer *plate;
static Layer *overlay;
static GBitmap *bitmaps[bitmaps_length];

static uint8_t prev_hour = 63, prev_min = 63;
```

git branch: master, index: √, working: 1≠, Comparing against : HEAD, ASCII, Line 1, Column 1 Spaces: 2 C++

main.c UNREGISTERED

main.c

```c

static void tick_handler(struct tm *t, TimeUnits units_changed) {
//  static char s_time_buffer[16];
//  strftime(s_time_buffer, sizeof(s_time_buffer), "%I:%M:%S", t);

//  APP_LOG(APP_LOG_LEVEL_DEBUG, "Uptime: %dh %dm %ds, flag: %d", t->tm_hour, t->tm_min, t->tm_sec, units_changed);

  // set clock time
//  uint8_t now_hour = (uint8_t)(*t).tm_hour > 12 ? (uint8_t)(*t).tm_hour - 12 : (uint8_t)(*t).tm_hour;
  int now_hour = t->tm_hour > 12 ? t->tm_hour - 12 : t->tm_hour;  // used less memory than above
  int now_min = t->tm_min / 5;

  // condition for update
  if (prev_hour != (uint8_t)now_hour || prev_min != (uint8_t)now_min) {
    prev_hour = (uint8_t)now_hour;
    prev_min = (uint8_t)now_min;

    layer_mark_dirty(overlay);
  }
}

static void update_light_layer(Layer *layer, GContext *ctx) {
  // set metrics
  const uint8_t digit_w = 24, digit_h = 26;
  const uint8_t col1 = 2, col2 = 31, col3 = 60, col4 = 89, col5 = 118;
  const uint8_t row1 = 6, row2 = 39, row3 = 72, row4 = 104, row5 = 137;

  const GRect h1 = GRect(col2, row1, digit_w, digit_h);
  const GRect h2 = GRect(col1, row2, digit_w, digit_h);
  const GRect h3 = GRect(col4, row1, digit_w, digit_h);
  const GRect h4 = GRect(col5, row1, digit_w, digit_h);
  const GRect h5 = GRect(col3, row1, digit_w, digit_h * 2 + 7);
  const GRect h6 = GRect(col2, row2, digit_w * 2 + 5, digit_h);
  const GRect h7 = GRect(col4, row2, digit_w * 2 + 5, digit_h);
  const GRect h8 = GRect(col1, row3, digit_w * 2 + 5, digit_h);
```

git branch: master, index: ✓, working: 1≠, Comparing against : HEAD, ASCII, Line 35, Column 1 Spaces: 2 C++

main.c UNREGISTERED

main.c

```
55
56  static void update_light_layer(Layer *layer, GContext *ctx) {
57    // set metrics
58    const uint8_t digit_w = 24, digit_h = 26;
59    const uint8_t col1 = 2, col2 = 31, col3 = 60, col4 = 89, col5 = 118;
60    const uint8_t row1 = 6, row2 = 39, row3 = 72, row4 = 104, row5 = 137;
61
62    const GRect h1 = GRect(col2, row1, digit_w, digit_h);
63    const GRect h2 = GRect(col1, row2, digit_w, digit_h);
64    const GRect h3 = GRect(col4, row1, digit_w, digit_h);
65    const GRect h4 = GRect(col5, row1, digit_w, digit_h);
66    const GRect h5 = GRect(col3, row1, digit_w, digit_h * 2 + 7);
67    const GRect h6 = GRect(col2, row2, digit_w * 2 + 5, digit_h);
68    const GRect h7 = GRect(col4, row2, digit_w * 2 + 5, digit_h);
69    const GRect h8 = GRect(col1, row3, digit_w * 2 + 5, digit_h);
70    const GRect h9 = GRect(col3, row3, digit_w * 2 + 5, digit_h);
71    const GRect h10 = GRect(col1, row1, digit_w, digit_h);
72    const GRect hour_mark = GRect(col5, row3, digit_w, digit_h);
73
74    const GRect o_clock = GRect(col1, row4, digit_w * 2 + 5, digit_h);
75    const GRect m5 = GRect(col4, row5, digit_w, digit_h);
76    const GRect m10 = GRect(col5, row4, digit_w, digit_h);
77    const GRect m10_2 = GRect(col3, row5, digit_w, digit_h);
78    const GRect m20 = GRect(col3, row4, digit_w, digit_h);
79    const GRect m30 = GRect(col4, row4, digit_w, digit_h);
80    const GRect m40 = GRect(col1, row5, digit_w, digit_h);
81    const GRect m50 = GRect(col2, row5, digit_w, digit_h);
82    const GRect min_mark = GRect(col5, row5, digit_w, digit_h);
83
84    // show hours
85    switch (prev_hour) {
86      case 0:
87        graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h2], h2);
88        graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h10], h10);
89        graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
```

git branch: master, index: ✓, working: 1✚, Comparing against : HEAD, ASCII, Line 35, Column 1 Spaces: 2 C++

main.c UNREGISTERED

main.c

```
83
84  // show hours
85  switch (prev_hour) {
86    case 0:
87      graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h2], h2);
88      graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h10], h10);
89      graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
90      break;
91    case 1:
92      graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h1], h1);
93      graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
94      break;
95    case 2:
96      graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h2], h2);
97      graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
98      break;
99    case 3:
100     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h3], h3);
101     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
102     break;
103   case 4:
104     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h4], h4);
105     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
106     break;
107   case 5:
108     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h5], h5);
109     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
110     break;
111   case 6:
112     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h6], h6);
113     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
114     break;
115   case 7:
116     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h7], h7);
117     graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
```

git branch: master, index: √, working: 1✦, Comparing against : HEAD, ASCII, Line 35, Column 1 Spaces: 2 C++

```
    break;
  case 12:
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h2], h2);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_h10], h10);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_hour_mark], hour_mark);
    break;
  default:
    break;
}


// show minutes
switch (prev_min) {
  case 0:
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_o_clock], o_clock);
    break;
  case 1:
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_m5], m5);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_min_mark], min_mark);
    break;
  case 2:
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_m10], m10);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_min_mark], min_mark);
    break;
  case 3:
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_m10], m10);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_m5], m5);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_min_mark], min_mark);
    break;
  case 4:
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_m20], m20);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_m10], m10_2);
    graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_min_mark], min_mark);
    break;
  case 5:
```

main.c UNREGISTERED

main.c

```
208       case 12:
209         graphics_draw_bitmap_in_rect(ctx, bitmaps[bitmap_o_clock], o_clock);
210         break;
211       default:
212         break;
213     }
214   }
215
216   static void load_layers(Layer *root_layer) {
217     GRect bounds = layer_get_bounds(root_layer);
218
219     // set plate
220     plate = bitmap_layer_create(bounds);
221     overlay = layer_create(bounds);
222
223     bitmap_layer_set_bitmap(plate, bitmaps[bitmap_plate]);
224
225     layer_add_child(root_layer, bitmap_layer_get_layer(plate));
226     layer_add_child(root_layer, overlay);
227
228     layer_set_update_proc(overlay, update_light_layer);
229   }
230
231   static void unload_layers() {
232     for (uint8_t i = 0; i < bitmaps_length; ++i) {
233       if (bitmaps[i]) gbitmap_destroy(bitmaps[i]);
234     }
235
236     layer_destroy(overlay);
237     bitmap_layer_destroy(plate);
238   }
239
240   static void window_load(Window *window) {
241     Layer *root_layer = window_get_root_layer(window);
```

git branch: master, index: √, working: 1✦, Comparing against : HEAD, ASCII, Line 35, Column 1 Spaces: 2 C++

main.c UNREGISTERED

main.c

```c
static void window_load(Window *window) {
  Layer *root_layer = window_get_root_layer(window);

  tick_timer_service_subscribe(HOUR_UNIT | MINUTE_UNIT, tick_handler);

  uint8_t resource_id = (uint8_t)RESOURCE_ID_HANNA_B;

  for (uint8_t i = 0; i < bitmaps_length; ++i) {
    bitmaps[i] = gbitmap_create_with_resource((uint8_t)(resource_id + i));
  }

  load_layers(root_layer);
}

static void window_unload(Window *window) {
  unload_layers();

  tick_timer_service_unsubscribe();
}

static void init(void) {
  window = window_create();
  window_set_window_handlers(window, (WindowHandlers) {
      .load = window_load,
      .unload = window_unload,
  });

  const bool animated = true;
  window_stack_push(window, animated);
}

static void deinit(void) {
  window_destroy(window);
}
```

main.c

```
260 static void init(void) {
261   window = window_create();
262   window_set_window_handlers(window, (WindowHandlers) {
263       .load = window_load,
264       .unload = window_unload,
265   });
266
267   const bool animated = true;
268   window_stack_push(window, animated);
269 }
270
271 static void deinit(void) {
272   window_destroy(window);
273 }
274
275 int main(void) {
276   init();
277 //  APP_LOG(APP_LOG_LEVEL_DEBUG, "Done initializing, pushed window: %p", window);
278
279   app_event_loop();
280
281   deinit();
282   return 0;
283 }
284
285
286
287
288
289
290
291
292
293
```


5:59 AM
Nation falls ...
6:43 AM
Sunrise
17°/7°
12:39 PM
Settings
Music
Notifications
Open app
Remove
Mute Weather

INTRODUCING PEBBLE TOOL 4.0

01 December 2015 | Katharine Berry | Freshly Baked | Comments

I am pleased to today announce that version 4.0-rc4 of the `pebble` tool is now available. The key new feature is a new paradigm for dealing with firmware and SDK versions. This makes it much easier to deal with differing SDK versions, or to test code on multiple (emulated) firmware versions.

A note: while the tool is now at version 4.0, the SDK, firmware and mobile apps will not be following. Pebble tool versioning is now completely independent of the rest of the Pebble ecosystem.

Managing SDKs

The pebble tool now manages SDKs for you, without you needing to download and install the entire SDK manually each time. The first time you need an SDK, the latest one will be automatically installed for you. After that, you can use the SDK operations that live under the `pebble sdk` subcommand.

To see a list of available SDKs, use `pebble sdk list`:

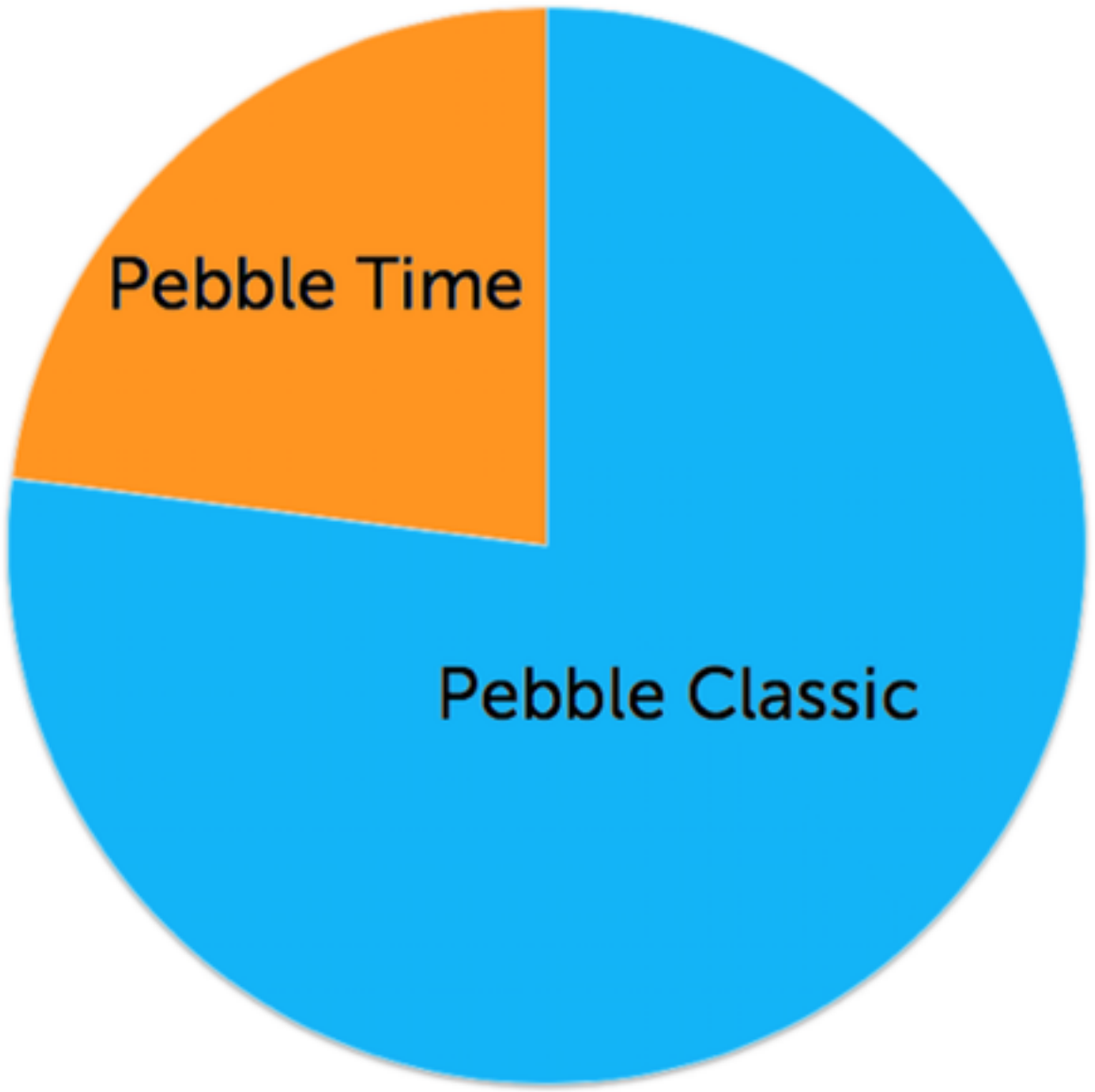
```
katharine@kbrmbp ~> pebble sdk list
Installed SDKs:
3.7 (active)

Available SDKs:
3.6.2
3.4
3.3
3.2.1
3.1
3.0
2.9
```

You can install any SDK using `pebble sdk install`, like so:

```
katharine@kbrmbp ~> pebble sdk install 3.6.2
Installing SDK...
Do you accept the Pebble Terms of Use and the Pebble Developer License? (y/n) y
Downloading...
100%[============================================================]   1.40 MB/s 0:00:01
Extracting...
Preparing virtualenv... (this may take a while)
Installing dependencies...
Done.
Installed.
```

```
Installed.
Done.
Installing dependencies...
Preparing virtualenv... (this may take a while)
Extracting...
100%[============================================================]   1.40 MB/s 0:00:01
Downloading...
Do you accept the Pebble Terms of Use and the Pebble Developer License? (y/n) y
Installing SDK...
katharine@kbrmbp ~> pebble sdk install 3.6.2
```

Maximize Userbase



Mr. Handy

페블 언어팩 개발 그룹 미스터 핸디를 소개합니다.

“You’re handy~!“

영화 매트릭스에서 트리니티가 열쇠맨에게 얘기한 대사가 기억에 남아 mr.Handy 라는 이름으로 팀을 만들었습니다. 언어팩 사용 지원 겸 페블 사용자 모임 채널입니다.

http://mr-handy.slack.com

누구나 가입할 수 있습니다.


Montréal
4:42
19°

아름답고 완성도 높은 한국어 언어팩 C83 을 소개합니다

한국어 언어팩 C83K 공개버전 다운로드

v1.6.1

Chinese support is coming to Pebble

March 17, 2015 Development No Comments
development, ecosystem, ios, kickstarter, news, notifications, Pebble, platform, smartwatch, software, watch, wearable technology, win

Chinese language support brings Simplified or Traditional Chinese to Pebble's apps notifications!

It's an exciting day for Chinese-reading Pebblers and fans. Starting now, Pebble's operating system and notifications will feature simplified and traditional Chinese language support, thanks to the Beta program going live today.

Activating Chinese language support for any Pebble is easy:

1. Make sure your Pebble's firmware has been updated to version 2.9.
2. Have the Pebble Android app (v2.3+) or the Pebble iOS app (v2.5+) installed on your smartphone.
3. Set your Pebble's language to Chinese (Simplified or Traditional) on your Pebble app in *Menu » Settings » Current Language » Selected Language.*
4. For more help updating your Pebble or Pebble apps for Chinese language integration, visit our help page here.

lp.getpebble.com/v1/languages


```
{"languages":[{"ISOLocal":"de_DE","file":"https://language-packs.s3.amazonaws.com/M1xu8Fi-
de_DE.pbl","firmware":"2.8.0","hardware":"v1_5","localName":"Deutsch","mobile":
{"name":"ios","version":"2.6.0"},"name":"German","version":1,"id":"547f360fdf4579140047174e"},
{"ISOLocal":"de_DE","file":"https://language-packs.s3.amazonaws.com/M1xu8Fi-
de_DE.pbl","firmware":"2.8.0","hardware":"v2_0","localName":"Deutsch","mobile":
{"name":"ios","version":"2.6.0"},"name":"German","version":1,"id":"547f362ab511f5130099cbe8"},
{"ISOLocal":"de_DE","file":"https://language-packs.s3.amazonaws.com/QxfUUVU-
de_DE.pbl","firmware":"2.8.0","hardware":"bb2","localName":"Deutsch","mobile":
{"name":"ios","version":"2.6.0"},"name":"German","version":1,"id":"547f38fedf4579140047174f"},
{"ISOLocal":"en_US","file":"https://language-packs.s3.amazonaws.com/yBE7uWg-
en_US.pbl","firmware":"2.8.0","hardware":"bb2","localName":"English","mobile":
{"name":"ios","version":"2.6.0"},"name":"English","version":1,"id":"547f6911df45791400471755"},
{"ISOLocal":"zh_CN","file":"https://language-packs.s3.amazonaws.com/IdlEZEQ-
zh_CN.pbl","firmware":"2.9.0","hardware":"v1_5","localName":"中文 (简体)","mobile":
{"name":"ios","version":"2.6.0"},"name":"Simplified Chinese","version":1,"id":"550874df17bfdb150071f8f"},
{"ISOLocal":"zh_CN","file":"https://language-packs.s3.amazonaws.com/lMG5CiT-
zh_CN.pbl","firmware":"2.9.0","hardware":"v2_0","localName":"中文 (简体)","mobile":
{"name":"ios","version":"2.6.0"},"name":"Simplified Chinese","version":1,"id":"550874f3a795921400f4d87a"},
{"ISOLocal":"zh_TW","file":"https://language-packs.s3.amazonaws.com/SutQecS-
zh_TW.pbl","firmware":"2.9.0","hardware":"v1_5","localName":"中文 (繁體)","mobile":
{"name":"ios","version":"2.6.0"},"name":"Traditional Chinese","version":1,"id":"5508750ea795921400f4d87b"},
{"ISOLocal":"zh_TW","file":"https://language-packs.s3.amazonaws.com/nwsTb46-
zh_TW.pbl","firmware":"2.9.0","hardware":"v2_0","localName":"中文 (繁體)","mobile":
{"name":"ios","version":"2.6.0"},"name":"Traditional Chinese","version":1,"id":"5508753bb175e91400099330"},
{"ISOLocal":"en_US","file":"https://language-packs.s3.amazonaws.com/X9Xx37z-Dn1N058-
en_US.pbl","firmware":"3.4.0","hardware":"snowy_dvt","localName":"English","mobile":
{"name":"ios","version":"2.6.0"},"name":"English","version":1,"id":"55ef3ea0d47f451400aba9d1"},
{"ISOLocal":"de_DE","file":"https://language-packs.s3.amazonaws.com/EkudbXX-
```

since the update, pebble time continuously restarts when I go to the quick launch menu..

MiRyoung Posts: 1 Member Pebbler

July 23 edited July 23 in General Discussion

Hi, I'm the pebble time user in Korea.
Thanks to local language pack developer, I'm using my pebble time in Korean.

I did the firmware update v3.2 today, and since then, my pebble time restarts when I go to the setting>quick launch menu.

I'm wondering this happens only to the pebble time which have korean language pack or the others.

How's it going on?
What about your pebble time, global pebblers?
appreciate your comments

Post edited by MiRyoung on July 23

Flag Quote

Comments

soomtong Posts: 1 Member Pebbler

July 23

Yeah, Me too. broken in there! :angry: also I did installed tha

Flag Quote

Destin Posts: 3 Member Pebbler

NEW DISCUSSION

Categories
Recent Discussions
Activity
My Drafts 1
Best Of...

Categories

All Categories 18.2K
Announcements and Updates 48
General Discussion 5.6K
Developer Discussion 7.2K
Official Developer News & 6
1.8K
1.8K
227
71
106
22
285
132
4.8K
Off Topic 412

PEBBLE SDK 3.2.1 - CHANGELOG

Release Date: August 12 2015

This is a hotfix release for SDK 3.2.

Known issues

- Taking screenshots from physical watches running firmware 3.2 or 3.2.1 usually doesn't work. We expect to fix this in firmware 3.3.

Changes to Firmware

- Apps compiled using SDK versions earlier than 3.2 which caused heap corruption but did not crash will now emit warnings instead of crashing. Apps compiled using SDK 3.2 or later will crash when a corrupt heap is detected.
- Resolved an issue causing emojis to display as boxes when replying to a notification using the large notification font.
- Resolved multiple issues caused by having unsupported language packs installed.

Heiko Behrens

hbehrens

JS on Pebble, Pebble in JS

What if you could run JS directly on the Pebble Smartwatch? What if you could run Pebble apps directly in the browser and program it with JS? There are a lot of interesting projects that deal with embedded devices and C/JS interoperability such as Espruino, JerryScript, Emscripten, or cheerp - let's do it all and see what happens to real firmware and real watches.

K[illegible] B[illegible]

pebble / **pebblejs**
forked from Meiguro/simplyjs

Watch 84 | Star 327 | Fork 190

Code | Issues 41 | Pull requests 3 | Pulse | Graphs

Program the Pebble with simply JavaScript

912 commits | **5** branches | **0** releases | **18** contributors

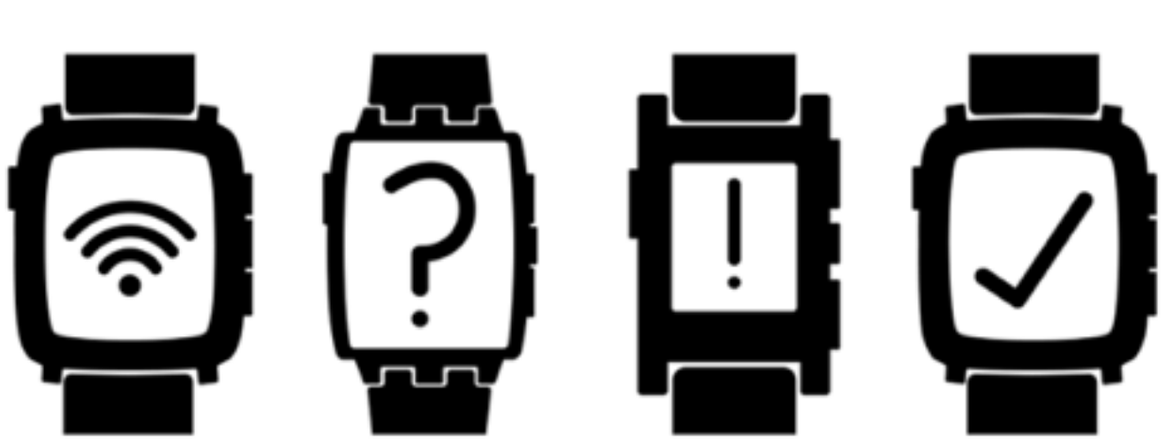
Branch: **master** | New pull request | New file | Find file | HTTPS | https://github.com/pebbl | Download ZIP

This branch is 777 commits ahead, 6 commits behind Meiguro:master. | Pull request | Compare

Meiguro Implement the voice module on aplite using only JavaScript ··· Latest commit `59fa17f` 3 days ago

resources	Change tile splash back to a white image of size 32x32	10 months ago
src	Implement the voice module on aplite using only JavaScript	3 days ago
waftools	Convert bitmap resources to png for older SDKs	5 days ago
.gitignore	Ignore compiled python files	5 days ago
.travis.yml	Change travis to use the new pebble tool for installing the SDK	5 days ago
LICENSE	Add the MIT license	2 years ago
README.md	Clarify that app.js xor app/ can exist	3 days ago
appinfo.json	Reformat appinfo.json following the configure appinfo output	5 days ago
doc.html	Added doc.html to provide local access to the documentation	2 years ago
wscript	Fix wscript formatting to follow PEP8	5 days ago

23
26
18
THU
10
80%
18
02




SAMSUNG'S GEAR S2
SMARTWATCH LOOKS AWESOME


android
wear


Pebble Developer Guide
Vol 1
0:01 / 10:24