
Pebble Smartwatch

— Watchface —

What is the Pebble smartwatch?

- Tech start up initially funded on kickstarter in 2012
- Considered the first smartwatch manufacturer
- Acquired by fitbit in late 2016, fitbit is continuing to support pebble software but not pebble hardware
- Pebble OS is expected to be featured in future fitbit products

2012



2013



2014



2015



2016



Two Methods For Creating Pebble OS Watchfaces:

C:

Advantages:

- Support for all generations of pebble
- Requires less pebble libraries

Disadvantages:

- Longer code
- Less intuitive syntax (in my opinion)
- Much more complicated to implement custom graphics

JavaScript:

Advantages:

- Easier and newer
- Works really well with new watches
- Relatively simple to implement vector graphics


Disadvantages:

- Not compatible with 1st generation watches
- Lots of external libraries
- Some features are still considered “beta”

C

- I created a digital watchface using C
- Text is easier than implementing graphics in C
- Watchfaces in C use window objects that refresh with a set frequency, typically every minute

CLOUDPEBBLE



FIRST WATCHFACE C

SETTINGS

TIMELINE (PREVIEW)

COMPILATION

DEPENDENCIES

GITHUB

APP SOURCE [ADD NEW](#)

main.c

PEBBLEKIT IS

DOCUMENTATION

```
1 #include <pebble.h>
2
3 static Window *s_main_window;
4 static TextLayer *s_time_layer;
5
6 static void update_time() {
7     // Get a tm structure
8     time_t temp = time(NULL);
9     struct tm *tick_time = localtime(&temp);
10
11     // Write the current hours and minutes into a buffer
12     static char s_buffer[8];
13     strftime(s_buffer, sizeof(s_buffer), clock_is_24h_style() ?
14             "%H:%M" : "%I:%M", tick_time);
15
16     // Display this time on the TextLayer
17     text_layer_set_text(s_time_layer, s_buffer);
18 }
19 static void tick_handler(struct tm *tick_time, TimeUnits units_changed) {
20     update_time();
21 }
22
23 static void main_window_load(Window *window) {
24     // Get information about the Window
25     Layer *window_layer = window_get_root_layer(window);
26     GRect bounds = layer_get_bounds(window_layer);
27
28     // Create the TextLayer with specific bounds
29     s_time_layer = text_layer_create(
30         GRect(0, PBL_IF_ROUND_ELSE(38, 52), bounds.size.w, 50));
31
32     // Improve the layout to be more like a watchface
33     text_layer_set_background_color(s_time_layer, GColorLiberty);
```

JavaScript

- I created an “analog” watch face with JavaScript, because it is simple to implement basic graphics
- Using an external library, events are created to change the position of the watch’s hands every time the the time changes (every second, minute, or hour)

