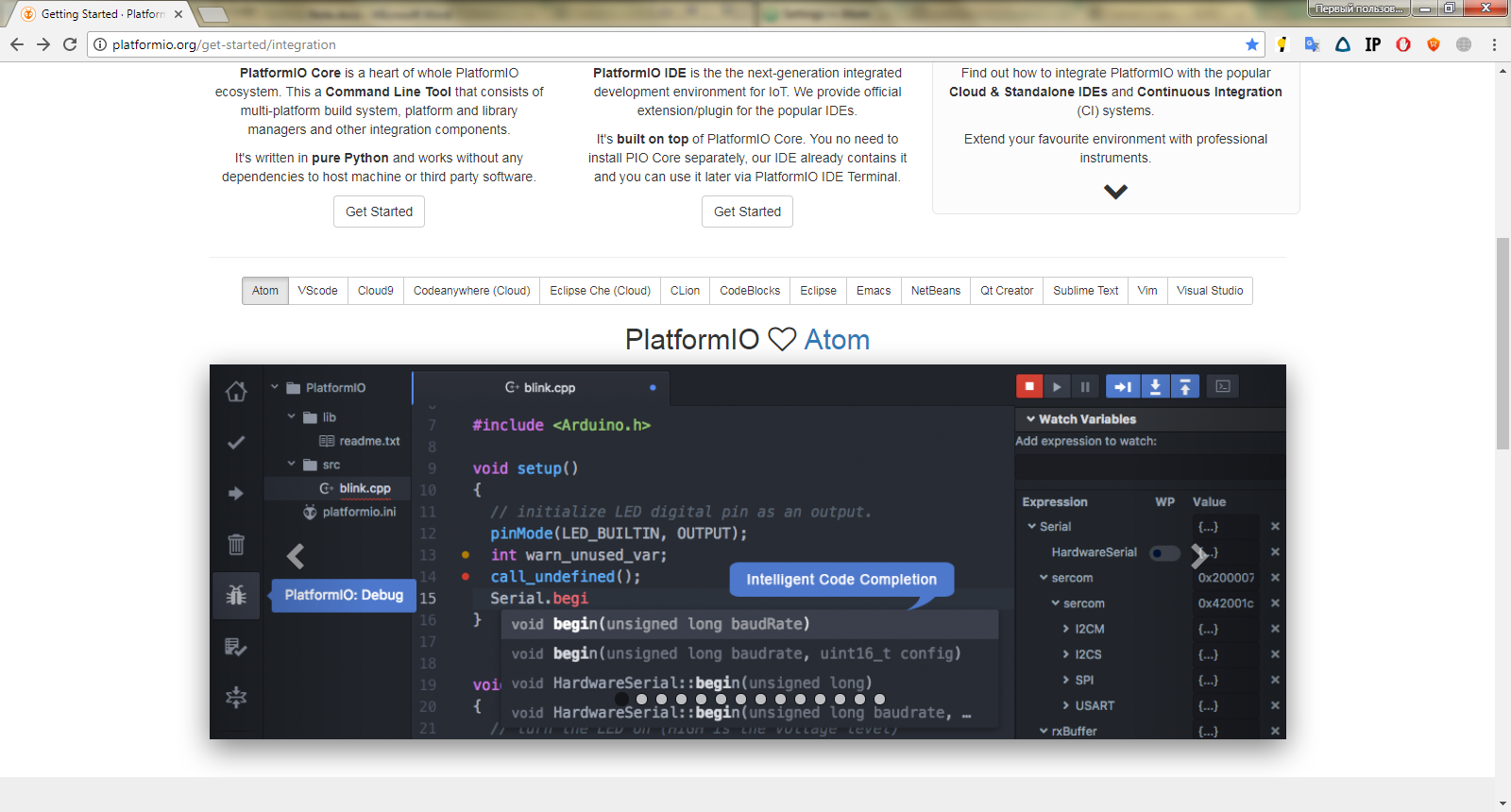
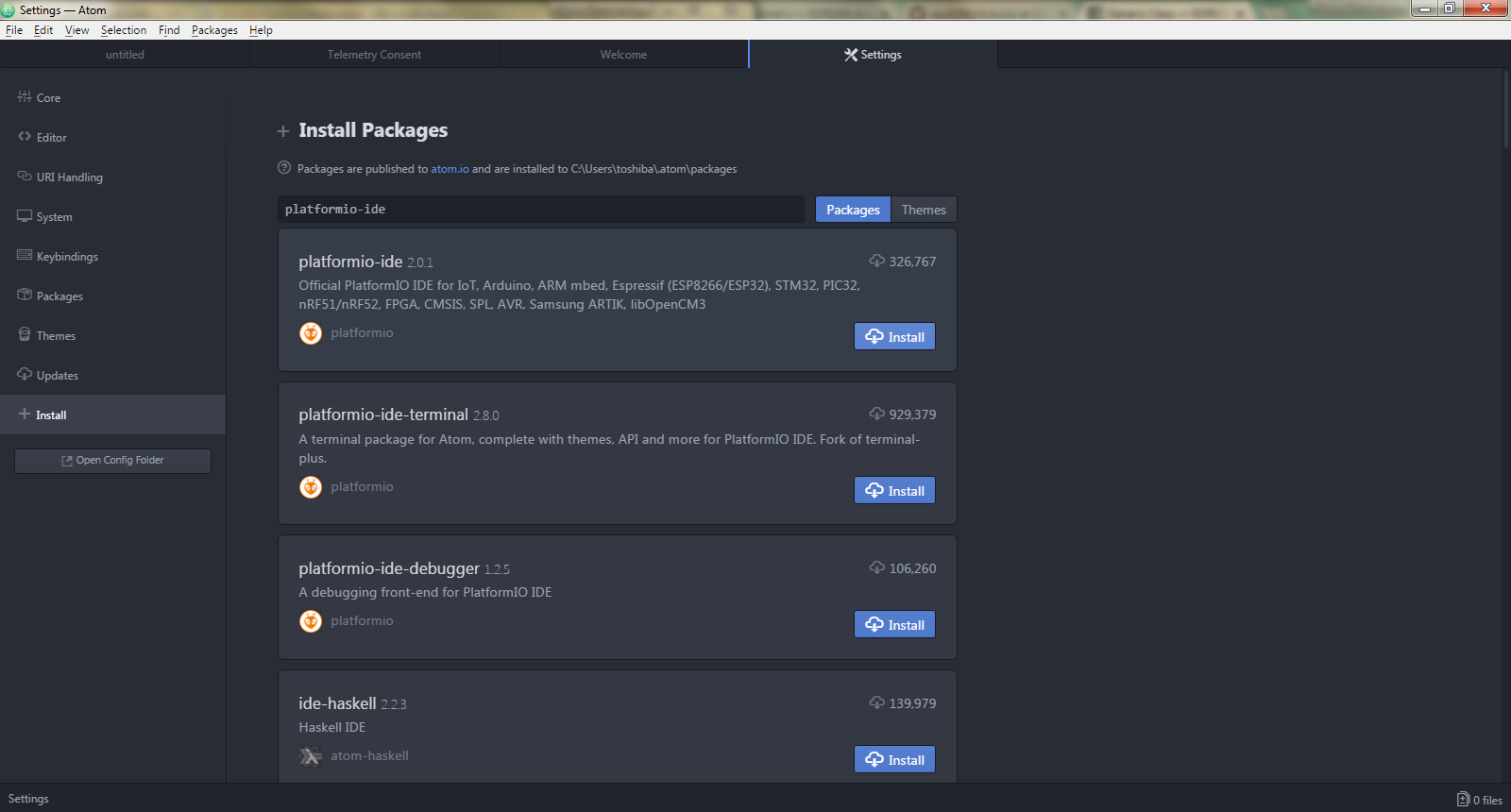
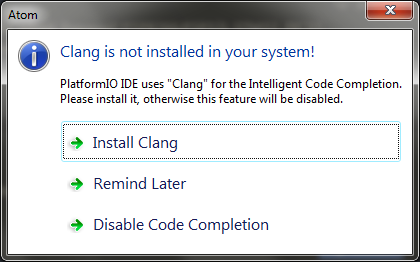
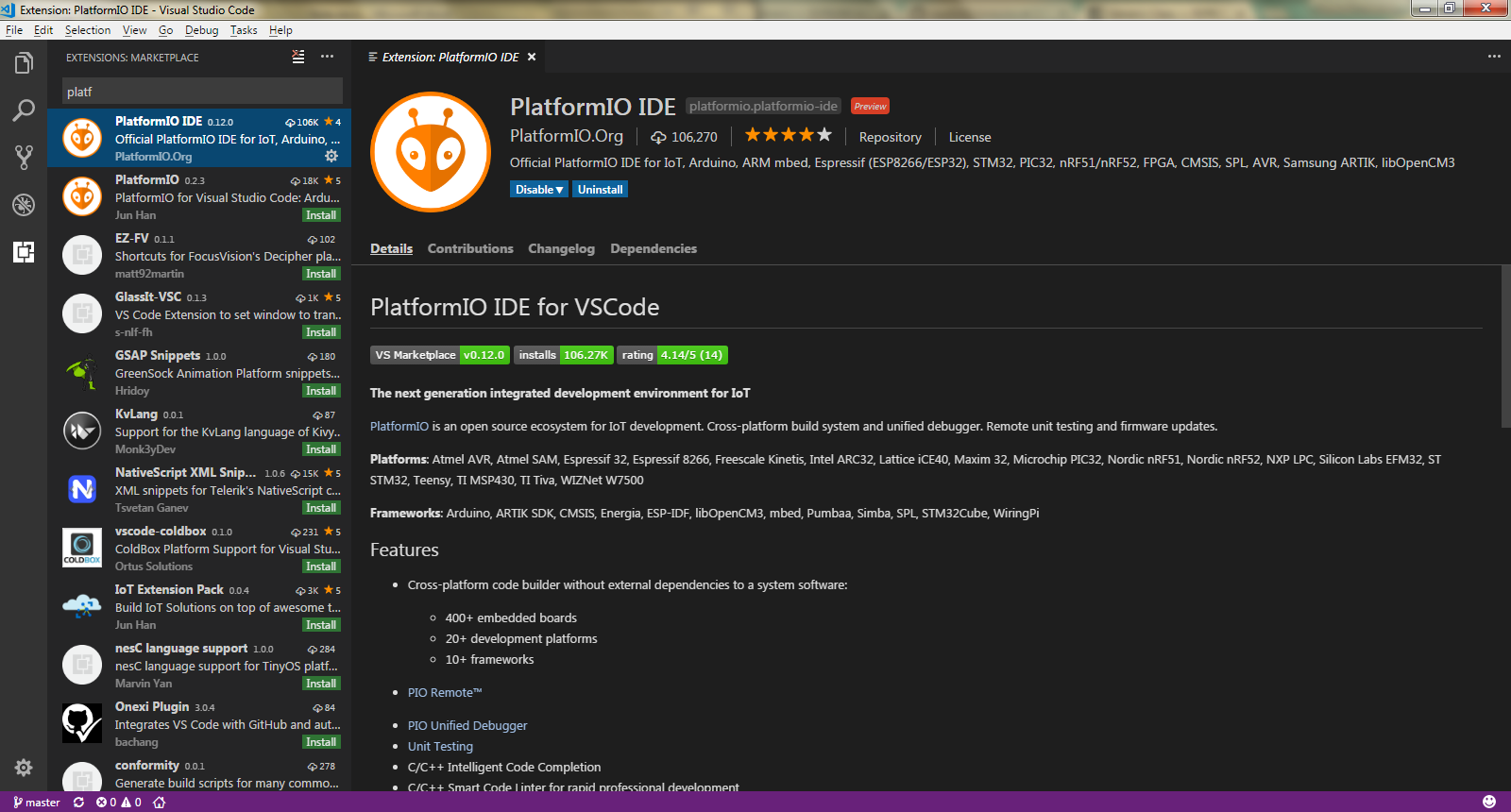
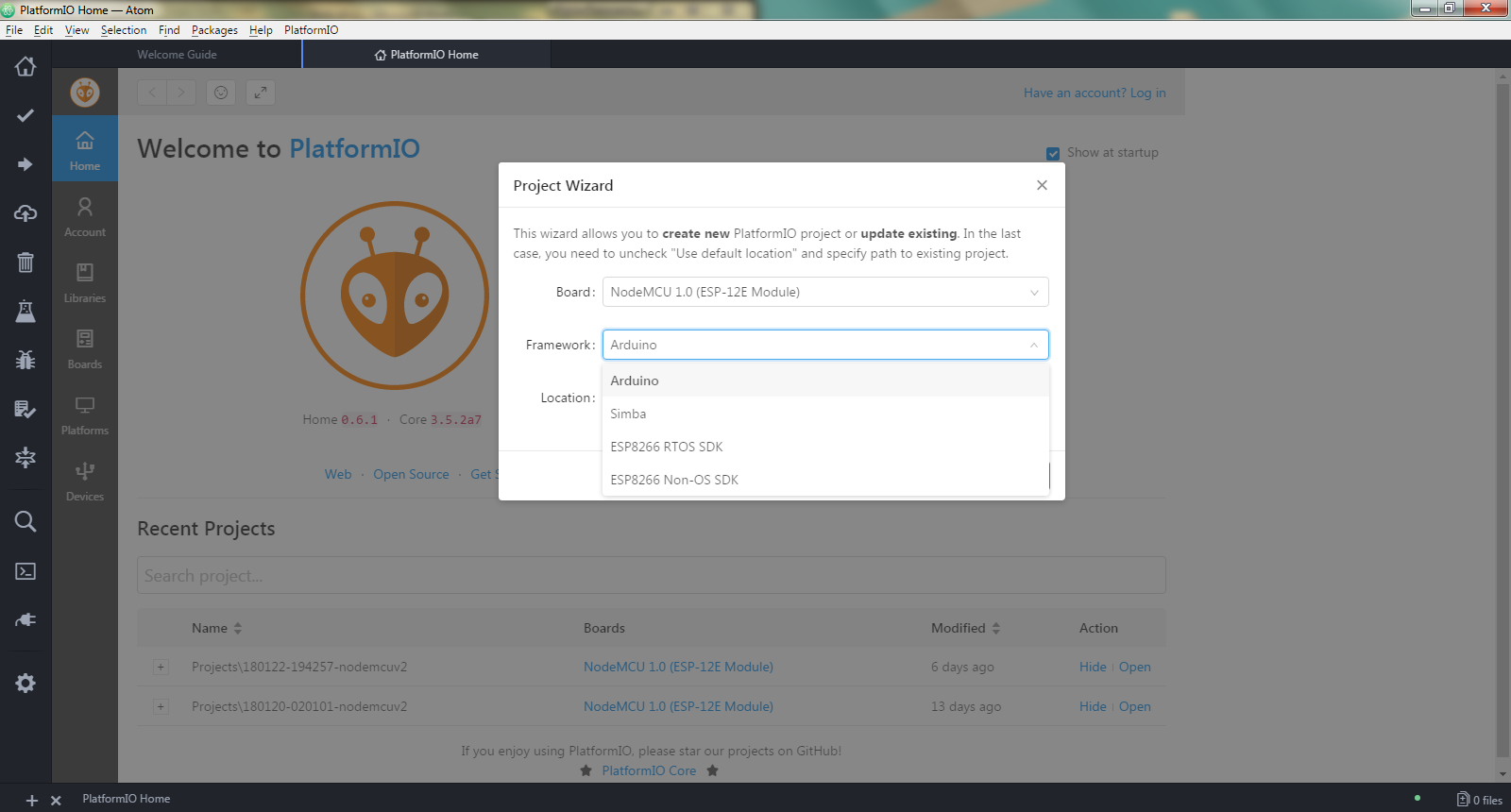
Покращимо собі умови з новим IDE, і з робимо це максимально «безболісно». Спочатку відвідаємо сайт [PlatformIO](http://platformio.org/get-started/integration) та оберемо для себе IDE в яку можна інтегрувати даний додаток. Виглядає це ось так:  
  
Далі обираємо для себе знайому або бажану IDE, і переходимо за посиланням на [англомовну інструкцію](http://docs.platformio.org/en/latest/ide.html), вона інтуїтивно зрозуміла тому проблем бути не повинно…  
В даному курсі ми оберемо [Atom IDE](https://atom.io/), через її переваги з графічним інтерфейсом для наших цілей. Але також розглянемо приклад роботи з консоллю адже це універсальне рішення.  
Отже завантаживши та встановивши Atom IDE (або інше) заходимо в *>File>Settings* , після чого з’являється панель налаштування на якій є кнопка *Install*. Шукаємо за запитом platformio-ide додаток і встановлюємо його.



В процесі установки Atom запросить встановити [Clang](http://docs.platformio.org/en/latest/ide/atom.html#ii-clang-for-intelligent-code-completion), тож підкоряємося (ну а VS Code потребує [Python 2.7](https://www.python.org/downloads/))  


  
Після встановлення перезавантажуємо IDE та створюємо новий проект у вікні PlatformIO. Обираємо свою плату та звичний нам Framework – Arduino.   
  
Бачимо нове меню проекту зліва та заходимо в теку src де вже є main.cpp файл, а не main.ino (при перенесені проекту з Arduino IDE треба змінити розширення на .cpp). Також бачимо відмінність нового проекту в цьому IDE від Arduino, що тут обов’язково повинна бути підключена бібліотека :

#include ***<Arduino.h>***

Це також треба враховувати при перенесені проекту. Тека lib призначена для зберігання не стандартних бібліотек які необхідні для проекту.  
Спробуємо написати програму Blink.cpp і завантажити її через це IDE. Для цього в Atom можна скористатися графічним інтерфейсом і навіть налаштувати COM-port, але в інших IDE можуть бути відсутні деякі функції в графічному інтерфейсі, тому [вчимося користуватися консоллю](http://docs.platformio.org/en/latest/userguide/index.html).

free(s)

malloc(s)

calloc(s)

realloc(p, s)

zalloc(s)

#include "Pins\_Arduino\_h"

static const uint8\_t SDA = 4;

static const uint8\_t SCL = 5;

static const uint8\_t LED\_BUILTIN = 16;

static const uint8\_t BUILTIN\_LED = 16;

static const uint8\_t D0 = 16;

static const uint8\_t D1 = 5;

static const uint8\_t D2 = 4;

static const uint8\_t D3 = 0;

static const uint8\_t D4 = 2;

static const uint8\_t D5 = 14;

static const uint8\_t D6 = 12;

static const uint8\_t D7 = 13;

static const uint8\_t D8 = 15;

static const uint8\_t D9 = 3;

static const uint8\_t D10 = 1;