

# iocafe

open source IoT/IO device communication

[home](#) [get involved](#) [services](#) [docs](#) [download](#) [blog](#)

## Configuring micro-controller WiFi with Android phone's LED flash

### Introduction

The “Gazerbeam” library enables configuring micro controller’s Wi-Fi network name (SSID) and password (PSK) using a LED flash of Android phone, etc.

- The phone needs APP to blink configuration information, the example Android Java application named “blinky”.
- A phototransistor is connected to a micro-controller with simple high pass filter/digitalization electronics.
- Micro-controller gets digital input signal from electronics. It is set up so that changes to the signal trigger interrupt. Gazerbeam library handles this and converts it to the message.
- The transferred data is encoded into the light signal using short and long intervals between the light on/off changes

0:00 / 1:08

*home video clip just to explain it*

### posts

[Installing NASA Core Flight System and OpenSatKit to Virtual Machine](#)

[PlatformIO is great](#)

[Democar project](#)

[ESP32: 1 MHz clock signal to output pin](#)

[Configuring micro-controller WiFi with Android phone's LED flash](#)

[IOCOM in a simple multiplayer game?](#)

[iocom secure network topology](#)

[testing development tools and secure communications](#)

### comments

[Marco Roda on Configuring micro-controller WiFi with Android phone's LED flash](#)

[Get Your Microcontroller Online at the Speed of Light – UpMyTech on](#)

# Android application

Android application is simple: To provide user form to fill in wifi network name, password and other optional configuration settings. When user fills these in and clicks “blink” button, the app starts sending this data repeatedly with LED flash until the “blink” button is clicked again.

[Configuring micro-controller WiFi with Android phone's LED flash](#)

pekkalehtikoski on [Configuring micro-controller WiFi with Android phone's LED flash](#)

pekkalehtikoski on [Configuring micro-controller WiFi with Android phone's LED flash](#)

[Get Your Microcontroller Online at the Speed of Light | 3d print](#)  
.....errori ed esperienze, on [Configuring micro-controller WiFi with Android phone's LED flash](#)

## archived

[July 2020](#)

[May 2020](#)

[April 2020](#)

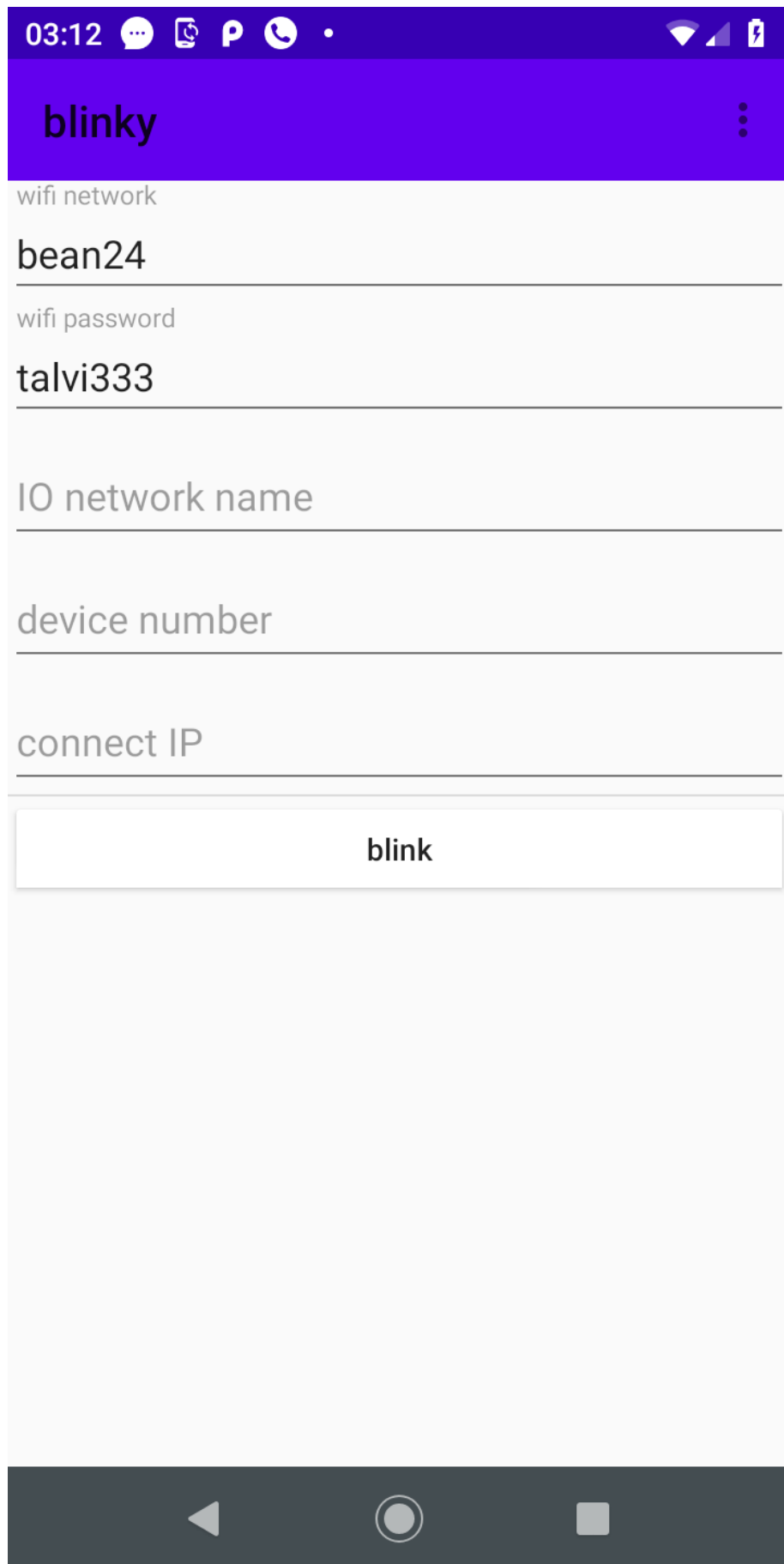
[March 2020](#)

[December 2019](#)

[September 2019](#)

## tags

cFS core Flight System Cosmos cubesat democar  
**esp32** espprog io-domain iocafe iocom mac  
NASA OBC OpenSatKit **platformio**  
programming flash release 2020 satellite secure  
network topology tls virtualbox **virtual**  
**machine** visual studio code

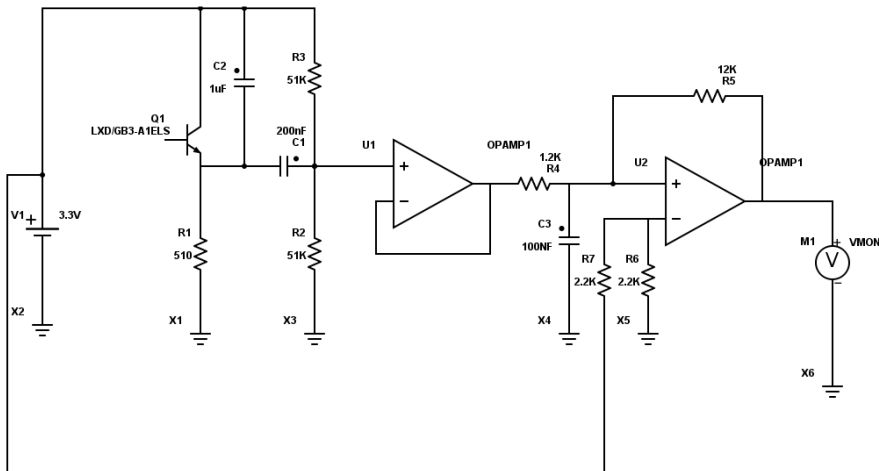


The android application's Java code is in  
[iocom/extensions/gazerbeam/examples/blinky](https://github.com/iocafe/iocom/extensions/gazerbeam/examples/blinky) directory.

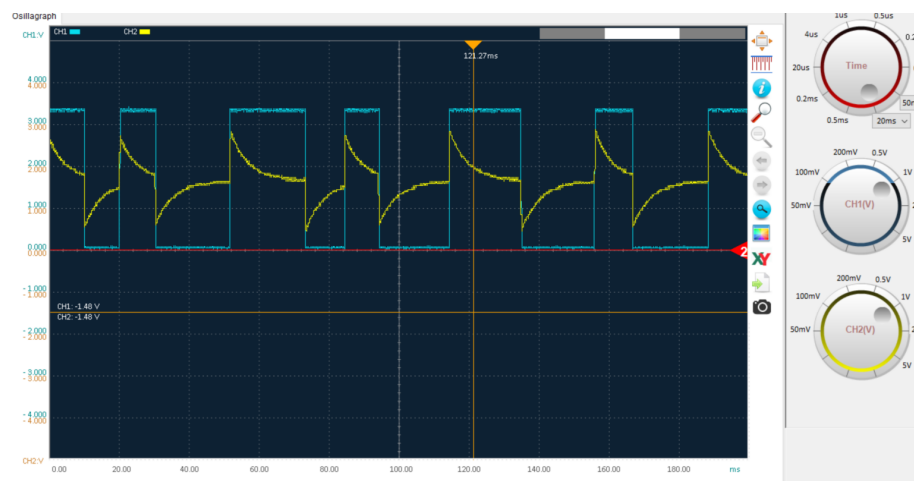
## Electronics

200318, updated 20.3.2020/pekka

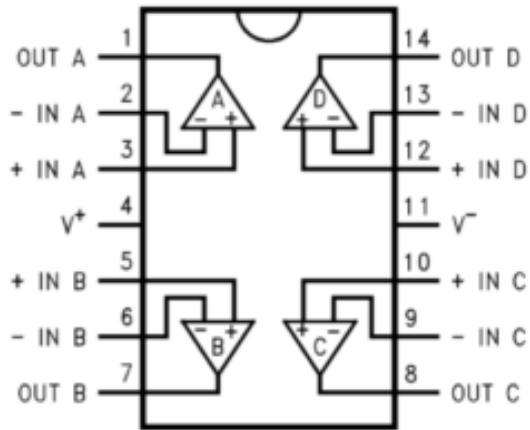
An LM6144 operational amplifier (U1 and U2) is used to high pass and to digitize the signal of LXD/GB3-A1ELS phototransistor (Q1).



The U2 is a comparator, which compares if the input voltage is less or greater than 1.65 V changes the voltage to 0V or 3.3V (roughly) Resistor R5 adds a threshold to change output state, to prevent output from filtering. In addition, there is mild low pass filtering by RC circuit C3/R4.



Yellow is a high pass filtered signal after the first OP-amp U1. The blue is the digitalized signal after the second OP-amp U2



**Figure 3. 14-Pin PDIP/SOIC  
Top View**

## Signal and data encoding

An Android phone can only turn flash on or off, the shortest pulse which can be reliably generated is approximately 10ms. This is enough, the amount of data to be transferred is small. Typically WiFi network name and password, and optionally some extra information.

### Zero and one bits

When sending Android's LED keeps on turning on and off continuously. What matters is how long LED stays in the same state: 10ms for "0" bit and 20 ms for "1" bit. So the time between

transitions matters, the direction of transition is ignored. It doesn't matter if LED switches from on to off or vice versa.

The Gazerbeam receiver is not very time-critical, timing can be changed quite much and it should still work. Pulse length is recorded on fly and basic rule is that "1" must be clearly longer than "0". Anyhow if taken very far from tested, the high pass filtering RC may need to be adjusted.

### From bits to messages

A message contains all data that the user entered and is repeated endlessly: The android application doesn't know when it is received by micro-controller.

- The message starts with 14 zero bits and then one bit, like "000000000000001".
- Then the 7 data bit for each data byte follow, least significant bit first.
- The data transferred is 7 bit ASCII with legitimate values from 1 to 127, it can never form nine consequent zeros and can be separated from beginning of message. Like "0101010".
- There is no message length. Previous message is processed when a new message begins, at that point the previous message is considered "ready".

- First three data bytes contain always checksum (MODBUS CRC), less significant byte first. First byte contains 6 least significant bits + 1, next byte 6 bits + 1 and the third byte 4 bits + 1.

### From bits to messages

The message starts with three-byte checksum. This needs to be generated by the Android code and is checked and stripped away by the Gazerbeam library.

Following the checksum there are 1 or more fields:

- A field starts with field id byte.
- Next byte is field data length, specifies the number of data bytes.
- N data bytes

## Using the library

### Data types and functions

The Gazerbeam receiver structure holds settings and the current state of monitoring phototransistor signal.

- Initialize gazerbeam structure and set pin to monitor.
- Poll for new received messages.

### The library source code

The source code is located in  
/coderoom/iocom/extensions/gazerbeam directory.

### Building the library

How to build the library depends on the target platform and tools used. Cmake builds are quite straight forward, there is CMakeLists.txt in the root directory, which can be opened for example in QT creator.

- Cmake: Either build project separately or include it as subproject within in your project
- ESP32/Arduino/Platform IO. Copy files. Configure platformio.ini.

### Include headers and link library with your project

The path iocom/extensions/gazerbeam should be added to your

project build. This allows including the library headers by  
“#include “gazerbeam.h”.

## Dependencies

The gazebeam depends on some libraries

- eosal – platform/operating system abstraction
- pins – portable access to IO pins
- iocom – some data types, not really much needed but dependency is there.

## HW support

- Tested on ESP32/Arduino/PlatformIO environment.
- To port to other hardware, the HW support needs to be implemented in eosal and pins libraries.

👤 pekkalehtikoski 📅 March 19, 2020 📁 technical notes

💬 7 Comments

← IOCOM in a simple multiplayer game?

ESP32: 1 MHz clock signal to output pin →

# 7 thoughts on “Configuring micro-controller WiFi with Android phone’s LED flash”

Pingback:[Get Your Microcontroller Online At The Speed Of Light – Hack A Day](#)

Pingback:[Get Your Microcontroller Online At The Speed Of Light | TECH, MUSIC & GAMING NEWS](#)

Pingback:[Get Your Microcontroller Online at the Speed of Light | 3d print .....errori ed esperienze,](#)

👤 pekkalehtikoski Post author

📅 April 28, 2020 at 16:48

🔗 Permalink



I am very happy and proud that this got an article in Hackaday:)

<https://hackaday.com/2020/04/28/get-your-microcontroller-online-at-the-speed-of-light/>

Hackaday has been a valuable source of information. Lately, instructions on TCD1304DG linear CCD helped me to get started and eventually it working. Earlier, an article about using GDB to debug Arduino code on STM32 saved a lot of time.

↳ Reply

👤 pekkalehtikoski Post author

📅 April 28, 2020 at 20:29

🔗 Permalink

I learned that Electric Imp has a somewhat similar way to set up the WiFi network connection. Electric Imp packs wifi adapter and ARM microcontroller into something which looks like a SD card. It is a ready product configured by a blinking phone display, while the method described here is more like "DIY" and uses a phone's flashlight.

<https://hackaday.com/2012/09/04/hands-on-with-the-electric-imp/>

↳ Reply

Pingback:[Get Your Microcontroller Online at the Speed of Light – UpMyTech](#)

👤 Marco Roda

📅 April 29, 2020 at 02:58

🔗 Permalink

Great Work!! I was just searching for something that could configure my WiFi credentials into my Project, this might be it. 😊 Thanks a lot for sharing! Keep it up.

↩ Reply

## Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment

Name \*

Email \*

Website



Save my name, email, and website in this browser for the next time I comment.

Post Comment

---

Copyright © 2020 iocafe. Powered by WordPress. Theme: Spacious by ThemeGrill.

---