# gazerbeam library

Configure microcontroller's WiFi using Android phone's flash light.

18.3.2020/Pekka Lehtikoski

## Table of Contents

# 1. Introduction

200318, updated 20.3.2020/pekka

The “gazerbeam” library enables configuring microcontroller’s WiFi network name (SSID) and password (PSK) using LED flash of Android phone, etc.

- The phone needs APP to blink configuration information, example Android Java application named “blinky” is in iocom/extensions/gazerbeam/examples/blinky directory.
- A photo transistor is connected to microcontroller tough electronics for it to receive the signal.
- Microcontroller gets signal from electronics as digital input. It is set up so that changes to the signal trigger interrupt. Gazerbeam library handles this and converts it to message.
- The transferred data is encoded into light signal using short and long intervals between loght on/off changes

## 1.1 Android application - Blinky

200318, updated 20.3.2020/pekka

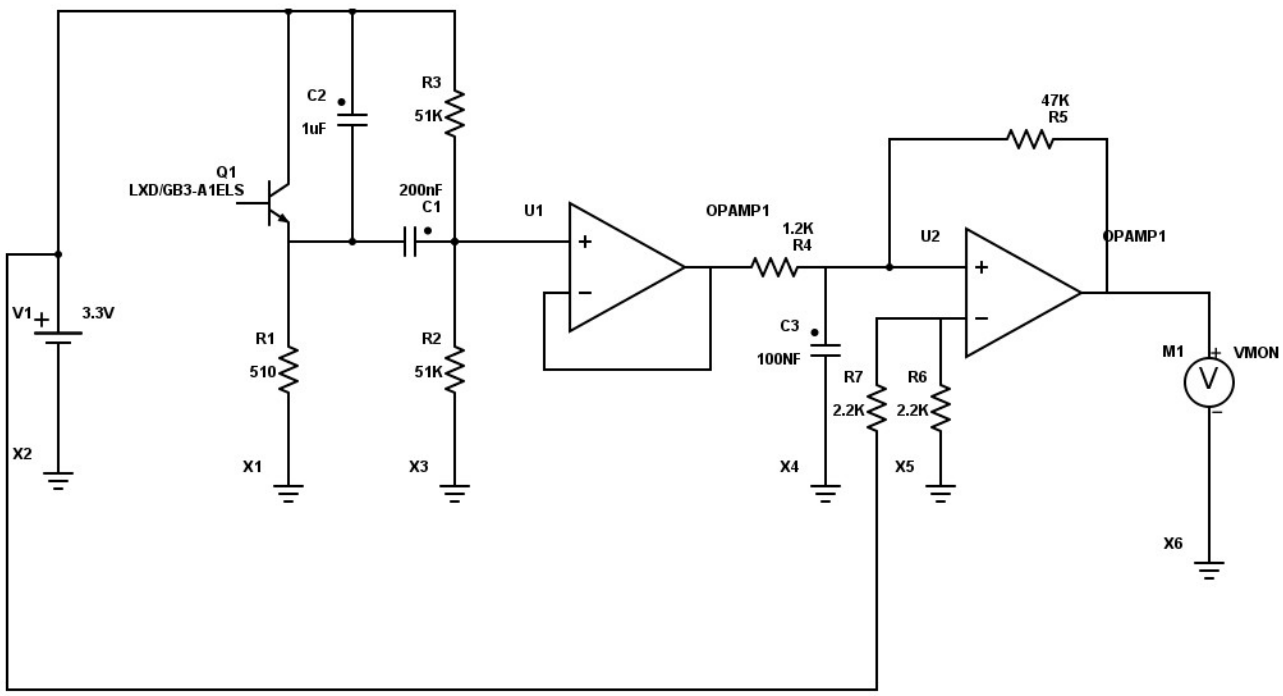
Android application is simple: To provide user form to fill in wifi network name, password and other optional configuration settings. When user fills these in and clicks “blink” button, the app starts sending this data repeatedly with LED flash until the “blink” button is clicked again.


The android application’s Java code is in iocom/extensions/gazerbeam/examples/blinky directory.

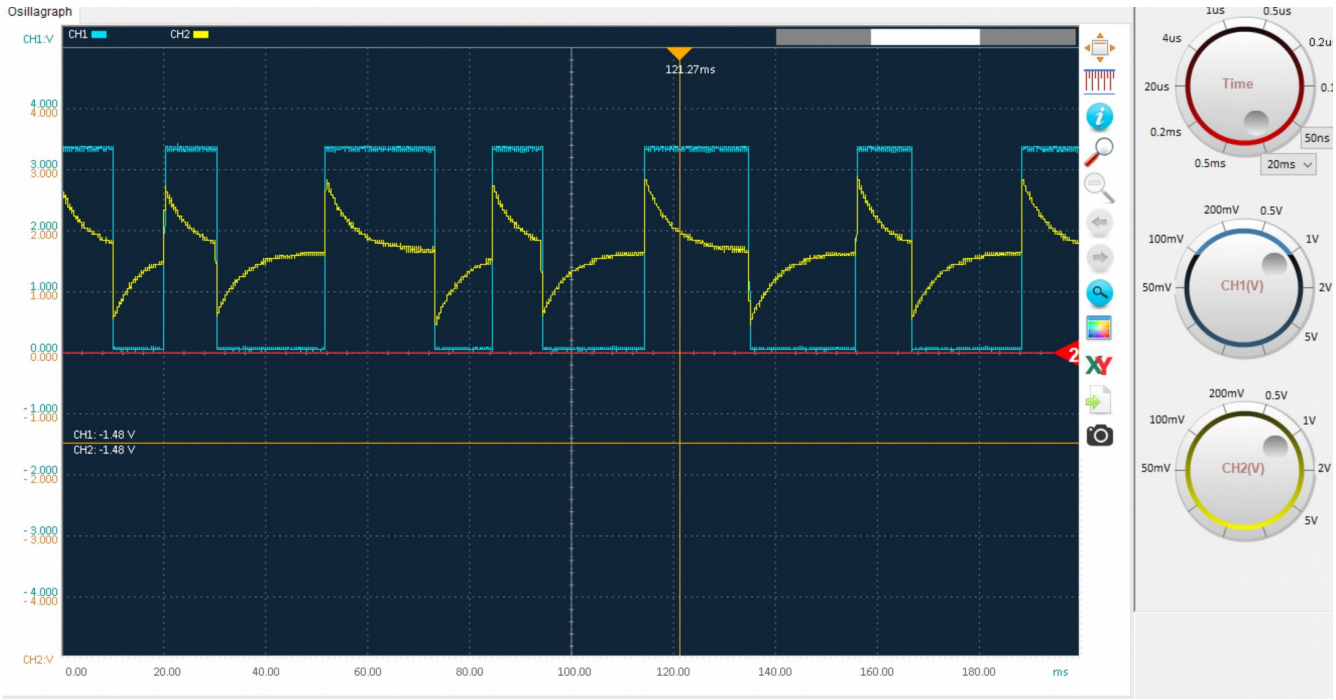
## 1.2 Electronics

200318, updated 20.3.2020/pekka

Fu




XXX




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

## 1.3 Signal encoding

200318, updated 20.3.2020/pekka

Standard Android phone can only turn flash on or off, and not control it's intensity. It is also relatively slow, shortest pulse which can be reliably generated is approximately 10ms. Luckily for this application this is enough,

amount of data to be transferred is small, typically WiFi network name and password, and optionally some extra information. Once wifi connection is establised, all data is transferred trough that.

When sending Android's LED keeps on turning on and off continuously. What matters is how long LED stays in same state: 10ms for "0" bit and 20 ms for "1" bit. So time between transitions matters, direction of transition is ignored. If doesn't matter if LED switches from on to off or vice versa.

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

- The message starts with 9 zero bits and then one bit, like "0000000001".
- Then the 8 data bit for each data byte follow, least significant bit first. An extra 1 bit is always appended to end. This data can never form nine consequent zeroes and can be separated from beginning of message. Like "010010101".
- There is no message length, nor end of message mark. Just when new message begins, the previous message is considered "ready".
- First two data bytes contain always checksum (MODBUS CRC), less significant byte first.

## 1.4 MIT license

190625, updated 27.6.2019/pekka