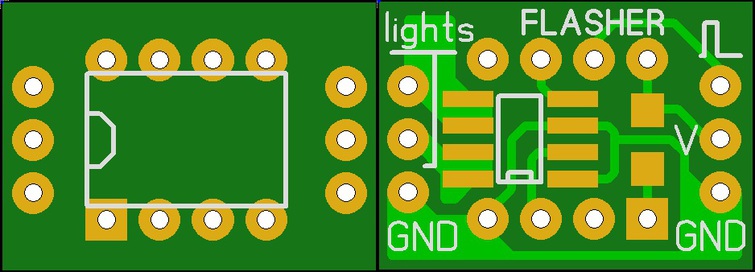
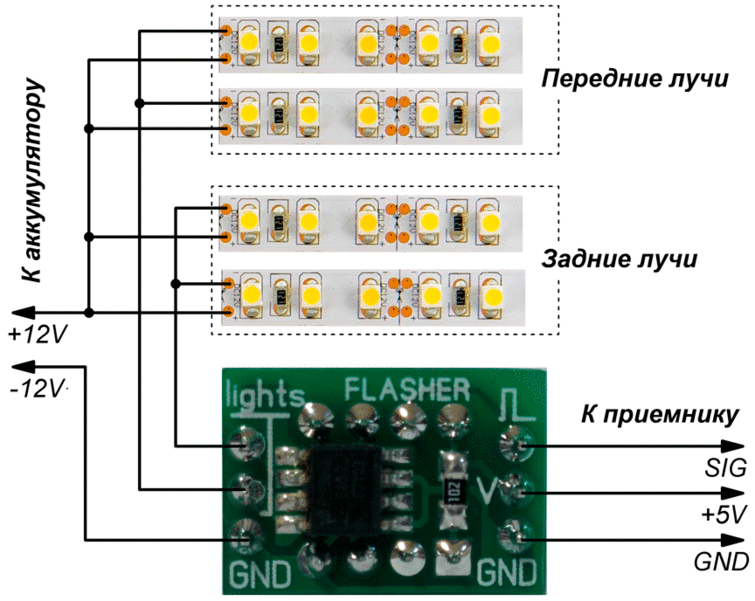
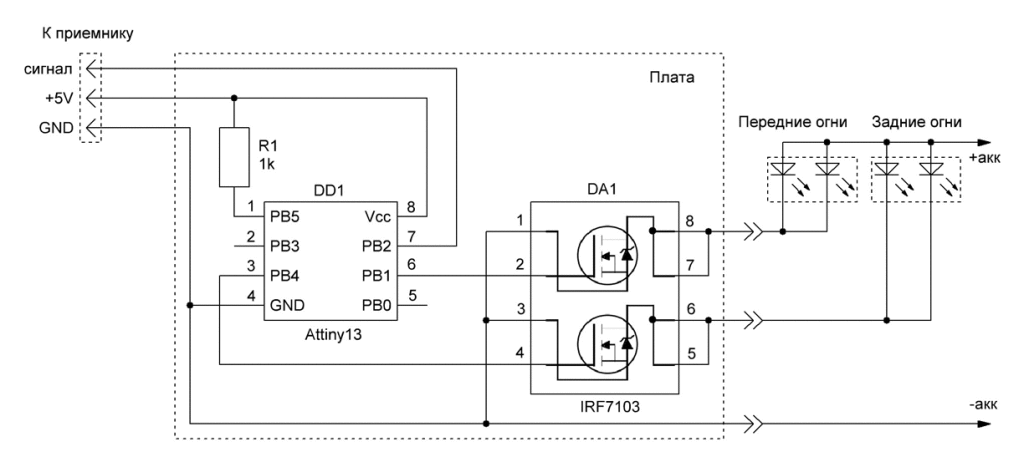


Плата предельно простая, односторонняя, дешевая и маленькая (всего 18х13мм). Для ее сборки потребуются следующие компоненты:

1. Микроконтроллер Attiny13A в DIP8 - 1шт
2. Транзисторная сборка IRF7103 в корпусе SO8 - 1шт
3. Резистор 0805-1кОм - 1шт
4. PLS-контакты - 6 штырьков
5. Колодка под корпус DIP8

Вот так выглядит сама плата (файл для скачивания будет в конце):  
[](http://www.parkflyer.ru/static/user_files/2016/1/11/FlasherBoard.1455217432977.jpg)  
Плата подключается одной стороной к приемнику (потребуется аппа с раздельными каналами), а другой стороной к подсветке. Все плюсы соединяются вместе, а минусы соединяются попарно и подводятся к контактам платы. Также наобходимо подключить к плате минус аккумулятора. Вот как это выгляит, если для подсветки используется обычная светодиодная лента:  
[[](http://www.parkflyer.ru/static/user_files/2016/1/11/CopterLightsWiring.1455217998114.gif)](http://www.parkflyer.ru/static/user_files/2016/1/11/CopterLightsWiring.1455217998114.gif)

Выходы платы рассчитаны на ток по 3А инапряжение по 50В на канал.

**Программирование**  
  
На данный момент наибольшей популярностью пользуется программа в которой реализовано четыре режима, задающихся с пульта (hex-файл в конце):

* огни выключены
* двойные вспышки
* попеременное мерцание (типа полицейской сирены)
* все огни включены

Но, если у вас есть программатор AVR-микроконтроллеров и минимальный опыт программирования МК, то вы сможете исправить эти режимы под себя.  
Вот исходный код программы:

Как вы знаете - вся программа начинается с функции main. В ней сначала происходит определение длительности сигнала, а затем, в зависимости от результата, запуск того или иного режима. То есть, например, строка if( mode == 1 ) doubleflash(); вызывает функцию, которая делает двойную вспышку.  
Вы можете легко поменять порядок режимов, выбрать другие или вообще прописать свои. Из готовых функций есть:

* always() - всегда включено
* flash() - однокраная вспышка
* doubleflash() - двойная вспышка
* change() - попеременное включение
* off() - полностью выключено
* siren() - мерцание сирены

Также можно пере прошить эту плату (используя те же функции) для управления фотоаппаратом, запалом, магнитом и т.д. Я специально выбрал контроллер в DIP-корпусе, чтобы его можно было легко достать и пере прошить.  
Чтобы скомпилировать вашу версию софта достаточно поставить AtmelStudio с официального сайта (бесплатно, без SMS и регистрации) и обзавестись программатором. Я рекомендую для таких целей иметь под рукой USBasp, либо программатор из Arduino.

/\*  
Flasher  
PB2 - PPM-signal  
PB1, PB4 - out  
CKSEL0=on; SUT0=on; CKDIV8=on; SPIEN=on (default fuses)  
\*/   
  
#include <avr/io.h>  
#define F\_CPU 1200000UL // 1.2 MHz  
#include <util/delay.h>  
  
uint8\_t mode = 0;  
  
void always() {  
PORTB |= ((1<<1)|(1<<4));  
}  
  
void flash() {  
PORTB |= ((1<<1)|(1<<4));  
\_delay\_ms(100);  
PORTB &= ~((1<<1)|(1<<4));  
\_delay\_ms(700);  
}  
  
void doubleflash() {  
PORTB |= ((1<<1)|(1<<4));  
\_delay\_ms(100);  
PORTB &= ~((1<<1)|(1<<4));  
\_delay\_ms(100);  
PORTB |= ((1<<1)|(1<<4));  
\_delay\_ms(100);  
PORTB &= ~((1<<1)|(1<<4));  
\_delay\_ms(700);  
}  
  
void change() {  
PORTB |= (1<<1);  
PORTB &= ~(1<<4);  
\_delay\_ms(500);  
PORTB |= (1<<4);  
PORTB &= ~(1<<1);  
\_delay\_ms(500);  
}  
  
void off() {  
PORTB &= ~(1<<4);  
PORTB &= ~(1<<1);  
}  
  
void siren() {  
PORTB |= (1<<1); \_delay\_ms(50);  
PORTB &= ~(1<<1); \_delay\_ms(50);  
PORTB |= (1<<1); \_delay\_ms(50);  
PORTB &= ~(1<<1); \_delay\_ms(50);  
PORTB |= (1<<1); \_delay\_ms(50);  
PORTB &= ~(1<<1); \_delay\_ms(50);  
  
PORTB |= (1<<4); \_delay\_ms(50);  
PORTB &= ~(1<<4); \_delay\_ms(50);  
PORTB |= (1<<4); \_delay\_ms(50);  
PORTB &= ~(1<<4); \_delay\_ms(50);  
PORTB |= (1<<4); \_delay\_ms(50);  
PORTB &= ~(1<<4); \_delay\_ms(50);  
}  
  
void HardwareInit() {  
PORTB |= (1<<2);  
DDRB &= (1<<2);  
PORTB &= ~((1<<1)|(1<<4));  
DDRB |= ((1<<1)|(1<<4));  
}  
  
int main(void)  
{  
HardwareInit();  
while(1)  
{  
mode = 0;  
while ( (PINB&(1<<2)) == (1<<2) ) {};  
while ( (PINB&(1<<2)) == 0b00000000 ) {};  
\_delay\_us(1200);  
if ( (PINB&(1<<2)) == (1<<2) ) mode++ ;  
\_delay\_us(200);  
if ( (PINB&(1<<2)) == (1<<2) ) mode++ ;  
\_delay\_us(200);  
if ( (PINB&(1<<2)) == (1<<2) ) mode++ ;  
if ( mode == 0 ) off();  
if ( mode == 1 ) doubleflash();  
if ( mode == 2 ) siren();  
if ( mode == 3 ) always();  
}  
}

