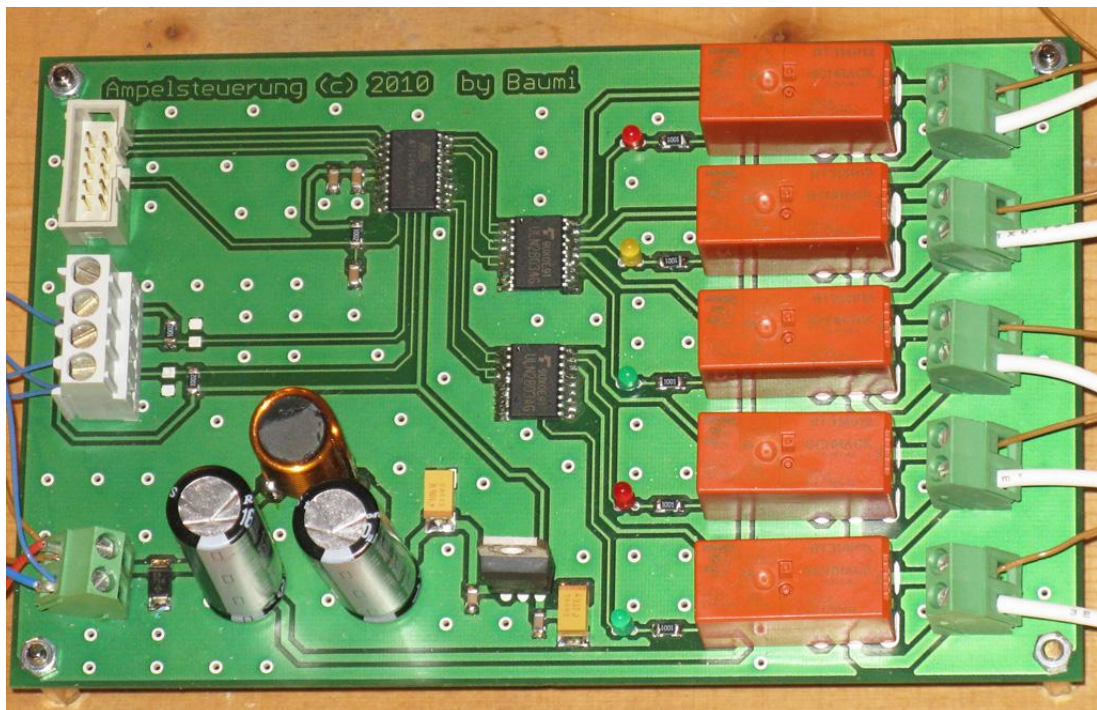


Projekt: Ampelsteuerung für die Kinderverkehrserziehung der Stadtpolizei Dornbirn



Ing. Bernhard Baumgartner
E-Mail: bernhard@baumgartners.at

Gesponsert von:



Inhalt

1 Funktionsbeschreibung:	3
2 Hauptkomponenten:	4
3 Beschaltung:	5
4 Verdrahtung:	5
5 Programmierung Funkmodul	6
6 Sourcecode	6
7 Schaltplan	8
8 Bestückungsplan	9
9 Stückliste:	10
10 Anlagen:	11

1 Funktionsbeschreibung:

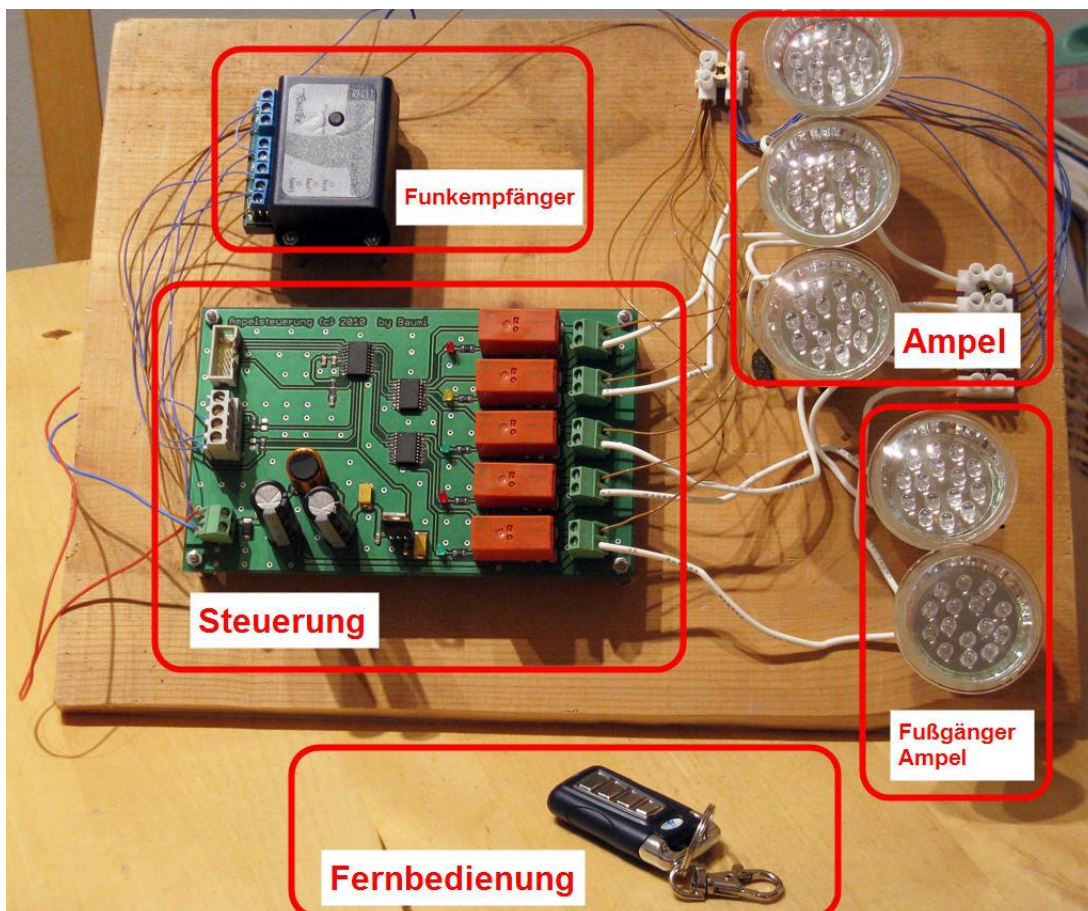
Die Ampelsteuerung simuliert eine Ampel an einem Fußgängerübergang. Es werden eine Ampel für Fahrzeuge (Rot, Gelb Grün) und eine Fußgängerampel (Rot & Grün). Die Umschaltung zwischen den einzelnen Phasen kann mittels Funkfernbedienung beziehungsweise einen Taster.

Der Ablauf der Ampelphasen ist wie folgt:

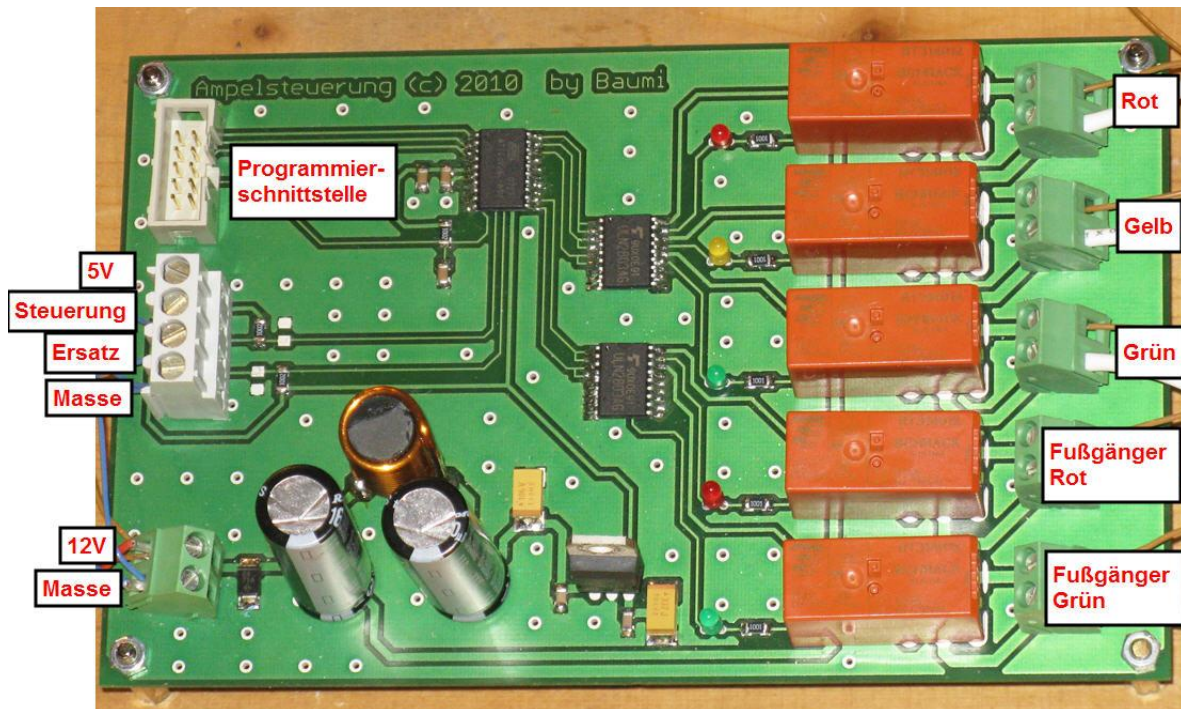
	Ampel	Fuß- gänger Ampel	Weiterschaltung in nächste Phase:
Phase 1			Tastendruck
Phase 2			Tastendruck
Phase 3		4x blinken	Automatisch
Phase 4			Tastendruck
Phase 5	rot gelb		Automatisch
Phase 6			Tastendruck
Phase 7	4x blinken		Automatisch
Phase 8			Automatisch zurück zu Phase 1

2 Hauptkomponenten:

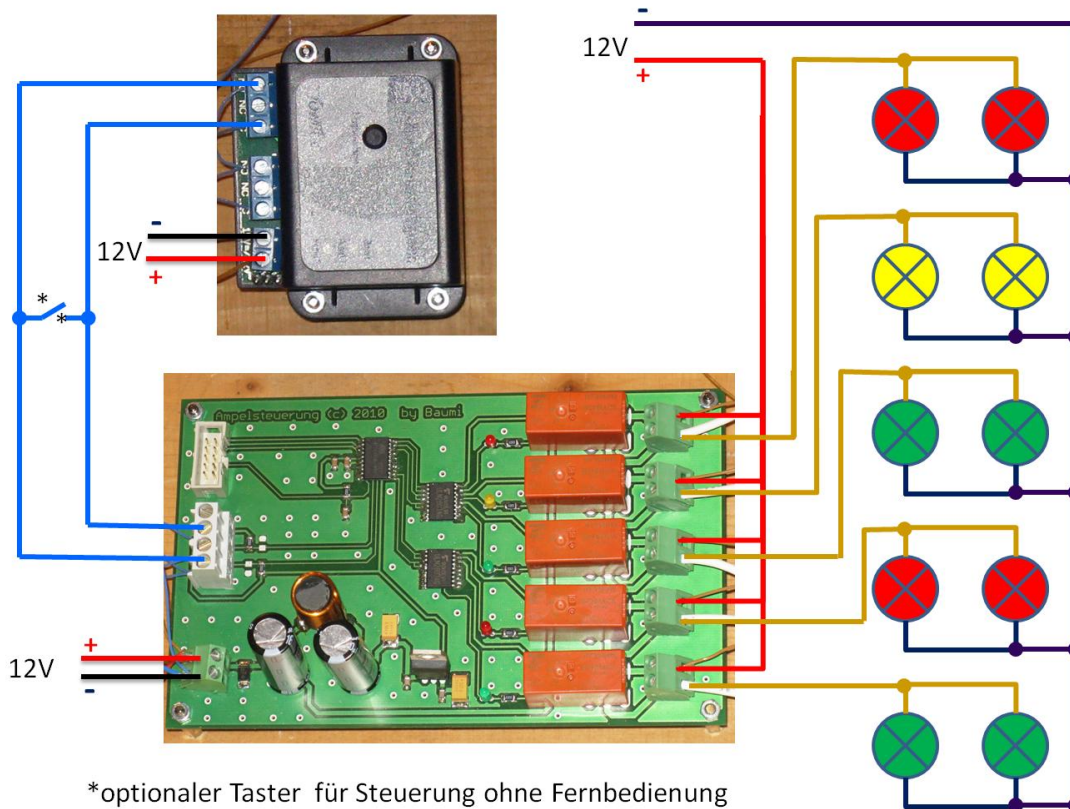
Beschreibung	Hersteller	Zusatzinfo
Ampelsteuerung	Eigenbau	
TowiTek Funkempfänger-Set	Conrad Elektronik	433 MHz Best. Nr: 191240-62
LED Lampen (rot, grün gelb)	Paulman	1W GU5,3 15°
Halterung für LED Lampen	Paulman	G4 GX5,3 GY 6,35 Art-Nr. 218.9



3 Beschaltung:



4 Verdrahtung:



Es können Leuchtmittel bis zu 240 VAC mit maximal 16A Laststrom verwendet werden.

5 Programmierung Funkmodul

Das Funkmodul wurde auf einen Impuls von 0.5 Sekunden programmiert. Informationen zur Programmierung des Funkmoduls befinden sich in der Anleitung des Moduls in der Anlage.

6 Source Code

```
//-----
// Project:   Ampel Stadtpolizei Dornbirn
// Author:    Bernhard Baumgartner
// Date:      16.08.2010 - 01.12.2010
// HW-rev:    Ampelsteuerung - Rev 1
// Prozessor: AT-Tiny 26
//-----

// Include Librarys:

#include <AVR/io.h>           // Definition der Ports
#include <stdlib.h>           // Standard Library
#include <stdio.h>            // Standard Input/Output Library
#ifndef F_CPU
#define F_CPU 1000000UL      /* Quarz mit 3.6864 Mhz */
#endif
#include <util/delay.h>      // Delay library

// define variables
char i,j;                    // Zählervariablen
int BLINK, GELB, ROTGELB, WAIT; // Variablen für Wartezeiten
char pinchange, old_pin, new_pin; // Hilfsvariablen für Pinänderungen

int main (void)
{
    DDRA = 0xff;             // programmiere Port A als Ausgang

    BLINK = 1000;            // Konstante Blinkdauer (Zeit für ON und OFF)
    WAIT = 500;              // Konstante Wartezeit nach Weiterschaltung
    GELB = 2500;             // Dauer der Gelb Phase
    ROTGELB = 2000;         // Dauer der ROT GELB Phase
    i = 7;                  // Initialisierung Zählvariable - ROT - ROT
    j = i;                  // Initialisierung Hilfszählvariable

    old_pin= 0x01;          // Initialisierung Old Pin
    new_pin = 0x01;         // Initialisierung New Pin
    pinchange = 0x00;      // Initialisiertun Pinchange

    while(1)
    {
        old_pin = new_pin;

        if (!(PINB & 0x40)) // Steuerpin einlesen
            new_pin = 0;
        else
            new_pin = 1;

        if ((new_pin != old_pin) && (new_pin == 0)) // testen auf fallende Flanke
        {
            ++i;
            pinchange = 1;
        }

        if ((i>7) || (i<1)) i=1; // Bereichsgrenzen sicherstellen
        j = i;
    }
}
```

```

switch (i)
{
    case 1:          PORTA = 0x81; // Auto Rot, Fussgaenger grün
                    break;

    case 2:          PORTA = 0x01;      // Auto Rot, Fussgänger grün blinken
                    _delay_ms(BLINK);
                    PORTA = 0x81;
                    _delay_ms(BLINK);
                    PORTA = 0x01;
                    _delay_ms(BLINK);
                    PORTA = 0x81;
                    _delay_ms(BLINK);
                    PORTA = 0x01;
                    _delay_ms(BLINK);
                    PORTA = 0x81;
                    _delay_ms(BLINK);
                    PORTA = 0x01;
                    _delay_ms(BLINK);
                    PORTA = 0x81;
                    _delay_ms(BLINK);
                    PORTA = 0x41; // Auto Rot, Fussgaenger rot
                    j = i+1;

                    break;

    case 3:          PORTA = 0x41; // Auto Rot, Fussgaenger rot
                    break;

    case 4:          PORTA = 0x43; // Auto Rot-Gelb, Fussgaenger rot
                    _delay_ms(ROTGELEB);
                    j = i+1;

                    break;

    case 5:          PORTA = 0x44; // Auto Grün, Fussgänger Rot
                    break;

    case 6:          PORTA = 0x40;      // Auto Grün blinken, Fussgänger Rot
                    _delay_ms(BLINK);
                    PORTA = 0x44;
                    _delay_ms(BLINK);
                    PORTA = 0x40;
                    _delay_ms(BLINK);
                    PORTA = 0x44;
                    _delay_ms(BLINK);
                    PORTA = 0x40;
                    _delay_ms(BLINK);
                    PORTA = 0x44;
                    _delay_ms(BLINK);
                    PORTA = 0x40;
                    _delay_ms(BLINK);
                    PORTA = 0x44;
                    _delay_ms(BLINK);
                    PORTA = 0x40;
                    _delay_ms(BLINK);
                    PORTA = 0x44;
                    _delay_ms(BLINK);
                    PORTA = 0x42;          // Auto Gelb , Fussgänger Rot
                    _delay_ms(GELB);
                    j= i+1;

                    break;

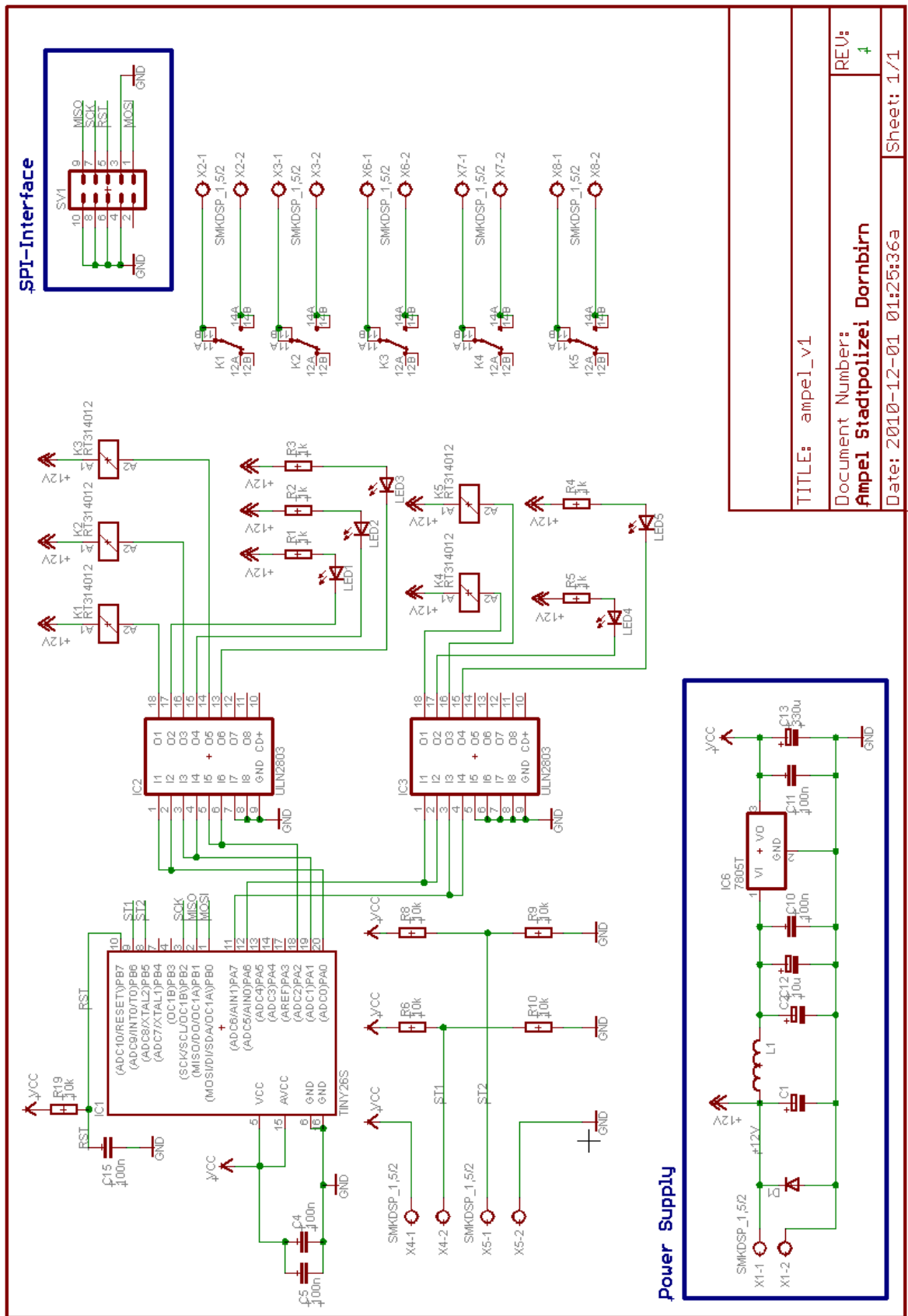
    case 7:          PORTA = 0x41; // Auto Rot, Fussgänger Rot
                    break;

};
i = j;
if (pinchange != 0)    // prüfen auf Weiterschaltung
{
    _delay_ms(WAIT);   // Steuerpin sperren
    pinchange = 0;
}

return 0;
}

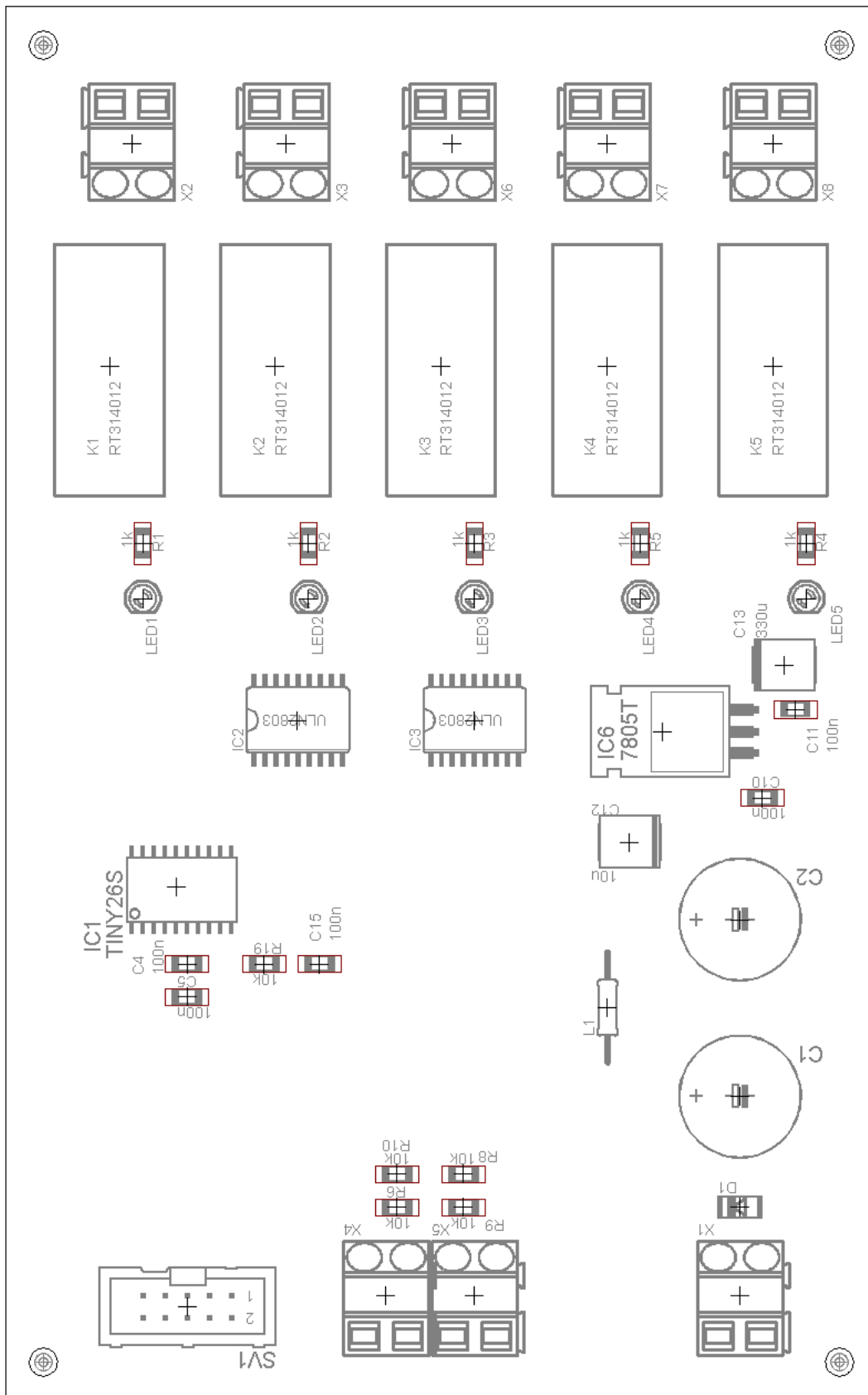
```

7 Schaltplan



TITLE: ampel_v1	REV: 4
Document Number: Ampel Stadtpolizei Dornbirn	
Date: 2010-12-01 01:25:36a	Sheet: 1/1

8 Bestückungsplan



9 Stückliste:

Part	Value	Device	Package	Library
C1	3900µ	CPOL-EUE5-13	E5-13	rcl
C2	3900µ	CPOL-EUE5-13	E5-13	rcl
C4	100n	C-EUC1206	C1206	rcl
C5	100n	C-EUC1206	C1206	rcl
C10	100n	C-EUC1206	C1206	rcl
C11	100n	C-EUC1206	C1206	rcl
C12	10u	CPOL-EUE/7260-38W	E/7260-38W	rcl
C13	330u	CPOL-EUE/7260-38W	E/7260-38W	rcl
C15	100n	C-EUC1206	C1206	rcl
D1		DIODE-MELF-MLL41	MELF-MLL41	diode
IC1	TINY26S	TINY26S	SO20L	atmel
IC2	ULN2803	ULN2803ASMD	SO18W	uln-udn_2
IC3	ULN2803	ULN2803ASMD	SO18W	uln-udn_2
IC6	7805T	7805T	TO220H	linear
K1	RT314012	RT314012	RT-C/O16A5	relay
K2	RT314012	RT314012	RT-C/O16A5	relay
K3	RT314012	RT314012	RT-C/O16A5	relay
K4	RT314012	RT314012	RT-C/O16A5	relay
K5	RT314012	RT314012	RT-C/O16A5	relay
L1		L-US0207/12	0207/12	rcl
LED1	LED Rot	LED3MM	LED3MM	led
LED2	LED Gelb	LED3MM	LED3MM	led
LED3	LED Grün	LED3MM	LED3MM	led
LED4	LED Rot	LED3MM	LED3MM	led
LED5	LED Grün	LED3MM	LED3MM	led
R1	1k	R-EU_R1206	R1206	rcl
R2	1k	R-EU_R1206	R1206	rcl
R3	1k	R-EU_R1206	R1206	rcl
R4	1k	R-EU_R1206	R1206	rcl
R5	1k	R-EU_R1206	R1206	rcl
R6	10k	R-EU_R1206	R1206	rcl
R8	10k	R-EU_R1206	R1206	rcl
R9	10k	R-EU_R1206	R1206	rcl
R10	10k	R-EU_R1206	R1206	rcl
R19	10k	R-EU_R1206	R1206	rcl
SV1	Stecker	ML10	ML10	con-harting-ml1
X1	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp
X2	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp
X3	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp
X4	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp
X5	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp
X6	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp
X7	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp
X8	SMKDSP_1,5/2	SMKDSP_1,5/2	SMKDSP_1,5/2	con-phoenix-smkdsp

10 Anlagen:

- Alle Projektdaten auf CD-ROM
- Bedienungsanleitung Funkempfänger-Set
- Schaltplan
- Bestückungsplan