Usring Dokumentation

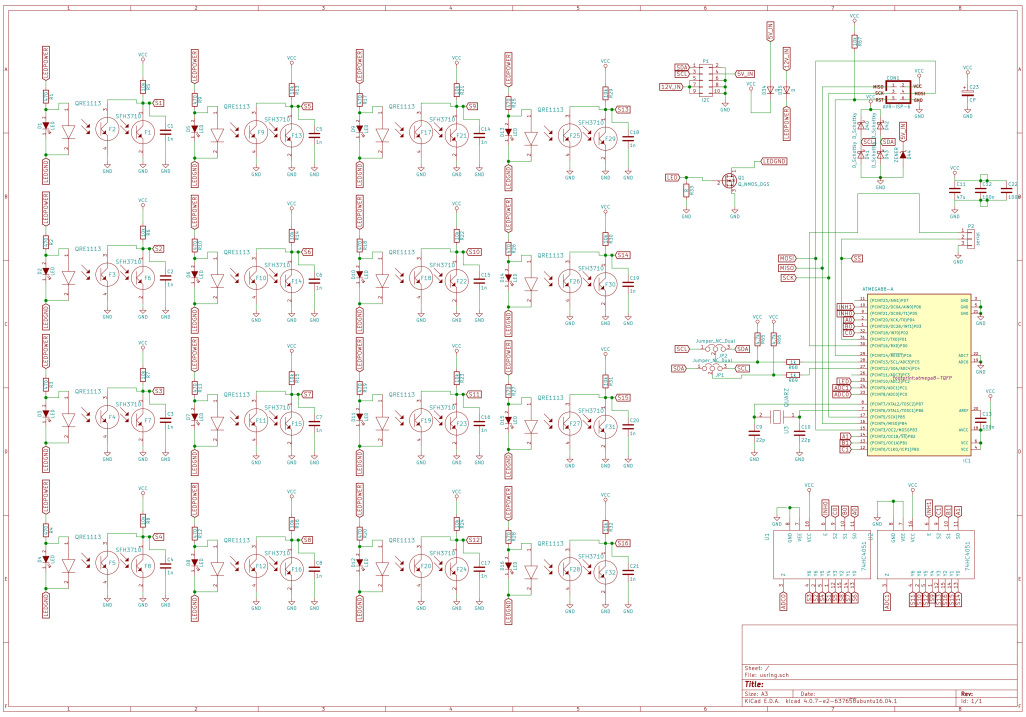
Alexander Ulbrich

7. März 2018

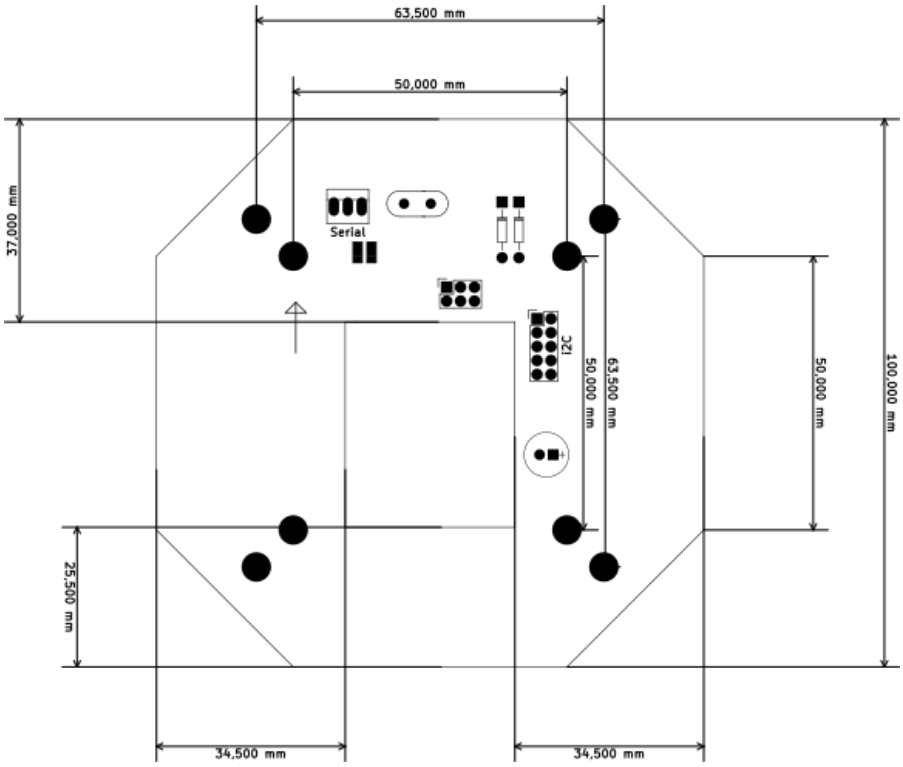
Inhaltsverzeichnis

1 Hardware

1.1 Schaltplan



1.2 Bemaßung



2 Software

2.1 usring Klasse

```
class Usring{
    Usring();
    void getDifferenceValue(int* valueBuffer);
    uint8_t getAnalogValue(uint8_t id);
    void setLed(uint8_t state);
    void setI2C(uint8_t id, uint8_t value);
    void setI2CData(uint8_t id, uint8_t size, uint8_t* data);
    uint8_t getI2C(uint8_t id);
    void print(const char* string);
    void print(int value);
};
```

2.2 Verwendung der usring Klasse

2.2.1 Konstruktor

Initialisert die gesammte Hardware.

```
Usring ring; // wie Goldboard gb;
```

2.2.2 Sensor getDifferenceValue

getDifferenceValue misst alle Sensoren einmal mit und einmal ohne LED Beleuchtung. Die Differenz der Messung wird im int Array valueBuffer gespeichert. valueBuffer muss die Laenge 16 haben.

```
ring.getDifferenceValue(int* valueBuffer);
```

2.2.3 Sensor getAnalogValue

getAnalogValue gibt den Analogwert des Sensors mit der Nummer id zurueck. Achtung! Es wird keine Differenz gemessen.

```
ring.getAnalogValue(uint8_t id);
```

2.2.4 Sensor setLed

setLed schaltet die LED an, wenn state ¿ 0 und aus wenn state = 0 ist.

```
ring.setLed(uint8_t state);
```

2.2.5 I2C setI2C

setI2C setzt den Wert value in das register mit der Nummer id. Der Wert einer bestimmten id kann direkt ueber i2c vom Master ausgelesen werden.

```
ring.setI2C(uint8_t id, uint8_t value);
```

2.2.6 I2C setI2CData

setI2CData setzt den Wert der Register mit der Nummer id bis zur Nummer id+size auf Werte in dem Array data mit der länge size.

```
ring.setI2CData(uint8_t id, uint8_t size, uint8_t* data);
```

Beispiel:

```
int x = 10;
ring.setI2CData(0, 2, (uint8_t*)&x);
```

2.2.7 I2C getI2C

getI2C gibt den Wert des Eingangs Registers mit der Nummer id zurueck. der I2C Master hat die Moeglichkeit diese Register zu setzen.

```
ring.getI2C(uint8_t id);
```

2.2.8 UART print

gibt die Zeichenkette string ueber Uart aus.

```
ring.print(const char* string);
```

gibt den Wert des Integeres value als ASCII ueber Uart aus.

```
ring.print(int value);
```

2.3 Beispielcode

```
#include "usring.h"

Usring ring;

int mittelwert[16];
int werte[16];
uint16_t ausgabe;

void kalibrieren()
{
    //mittelwerte auf 0 setzen
    for (int x = 0; x < 16; x++)
        mittelwert[x] = 0;

    //10 Messwerte Aufnehmen und den mittelwert Berechnen
    for (int x = 0; x < 10; x++)
    {
        ring.getDifferenceValue(werte);
        for (int i = 0; i < 16; i++)
            mittelwert[i] += werte[i];
        delay(10);
    }
    for (int x = 0; x < 16; x++)
        mittelwert[x] /= 10;
}

void messen()
{
    ring.getDifferenceValue(werte);
}
```

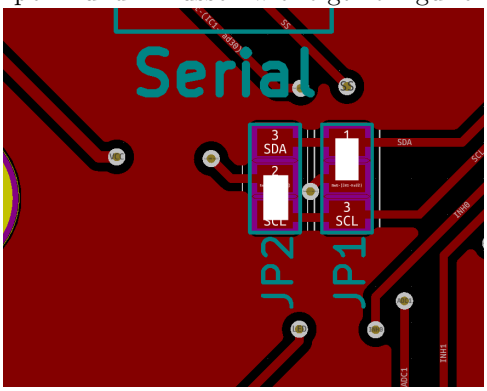
```
void berechnen()
{
    //Schwellwerte auswerten und
    //    Ergebnisse in 16 bit Typ Speichern
    ausgabe = 0;
    for (int x = 0; x < 16; x++)
    {
        // offset auf den mittelwert addieren
        if (werte[x] > mittelwert[x] + 10)
        {
            // setze das bit auf 1 wenn der sensor was sieht
            ausgabe |= (1<<x);
        }
    }
}


//daten in i2c Puffer Schreiben
void ausgeben()
{
    //16 bit Variable schreiben
    ring.setI2CData(0,2,(uint8_t*)&ausgabe);
    for (int x = 0; x < 16; x++)
        //analog werte schreiben
        ring.setI2C(x+2, werte[x]);
}

int main (void) {
    kalibrieren();
    while (1) {
        messen();
        berechnen();
        ausgeben();
    }
}
```

3 Bestellliste

Package	Anzahl	Wert
R_0805	16	470R
R_0805	3	1k
R_0805	17	10k
R_0805	2	4,7k
C_0805	4	100n
C_0805	2	22p
CP_Radial_D8.0mm_P2.50mm	1	100u
SFH3710	16	SFH3710
1206_LED	16	LED
Diode_throughole	2	D
SMD-shotkey	1	ZENER 5,1V
SOT23	1	Q_NMOS_DGS
QUARZ-HC49	1	QUARZ
Wannenstecker_2x05	1	I2C
Shotkey-SOD-123	4	D_Schottky
atmega8-TQFP	1	ATMEGA88-A
74HC4051-SO16	2	74HC4051

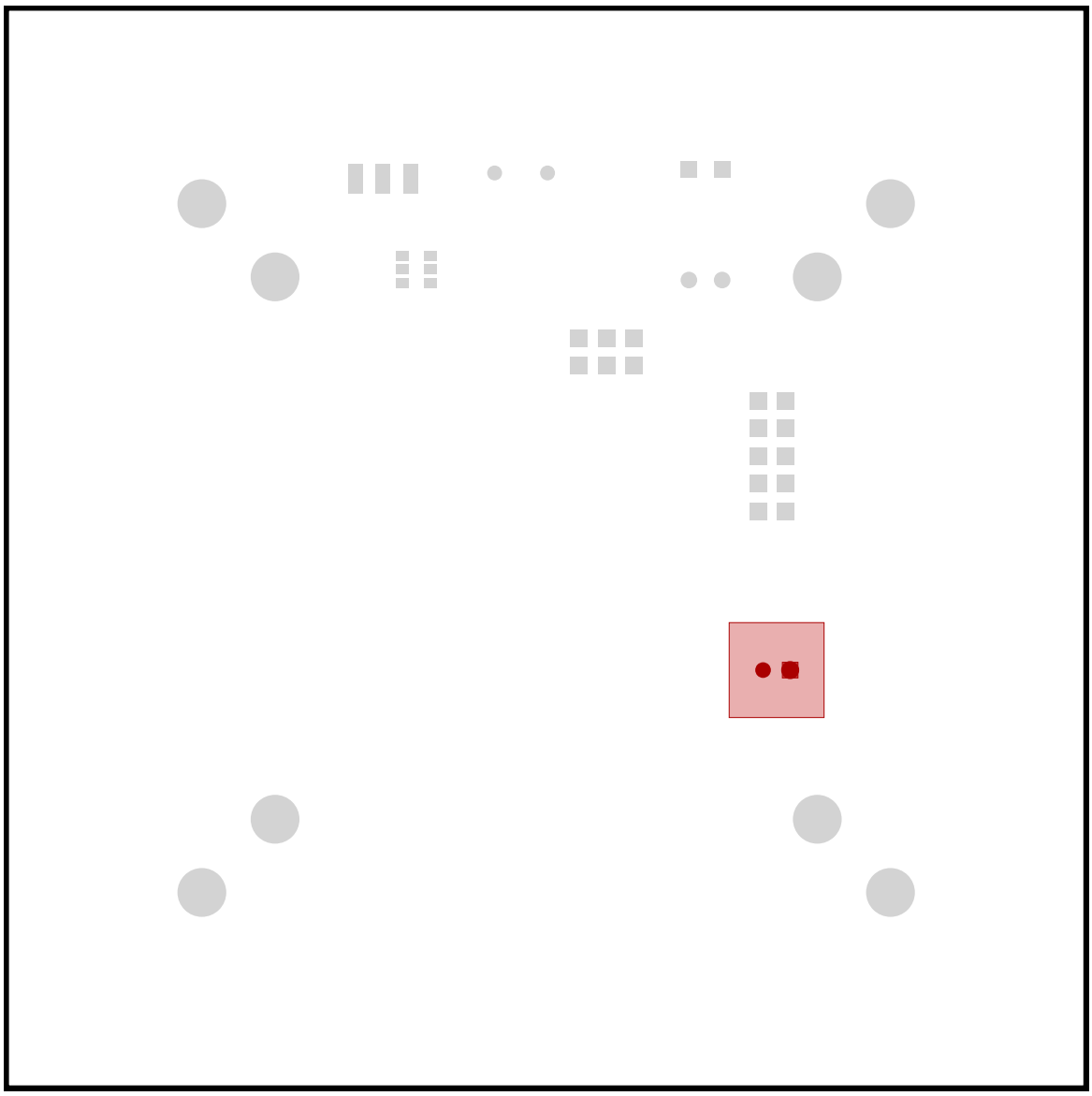
4 Bestückungsplan

Bei der Bestückung ist zu beachten, dass der Pin VEE mit GND verbunden werden muss (neben dran) die Verbindung mit VCC muss gekappt werden! Die Jumper 1 und 2 müssen wie folgt konfiguriert werden



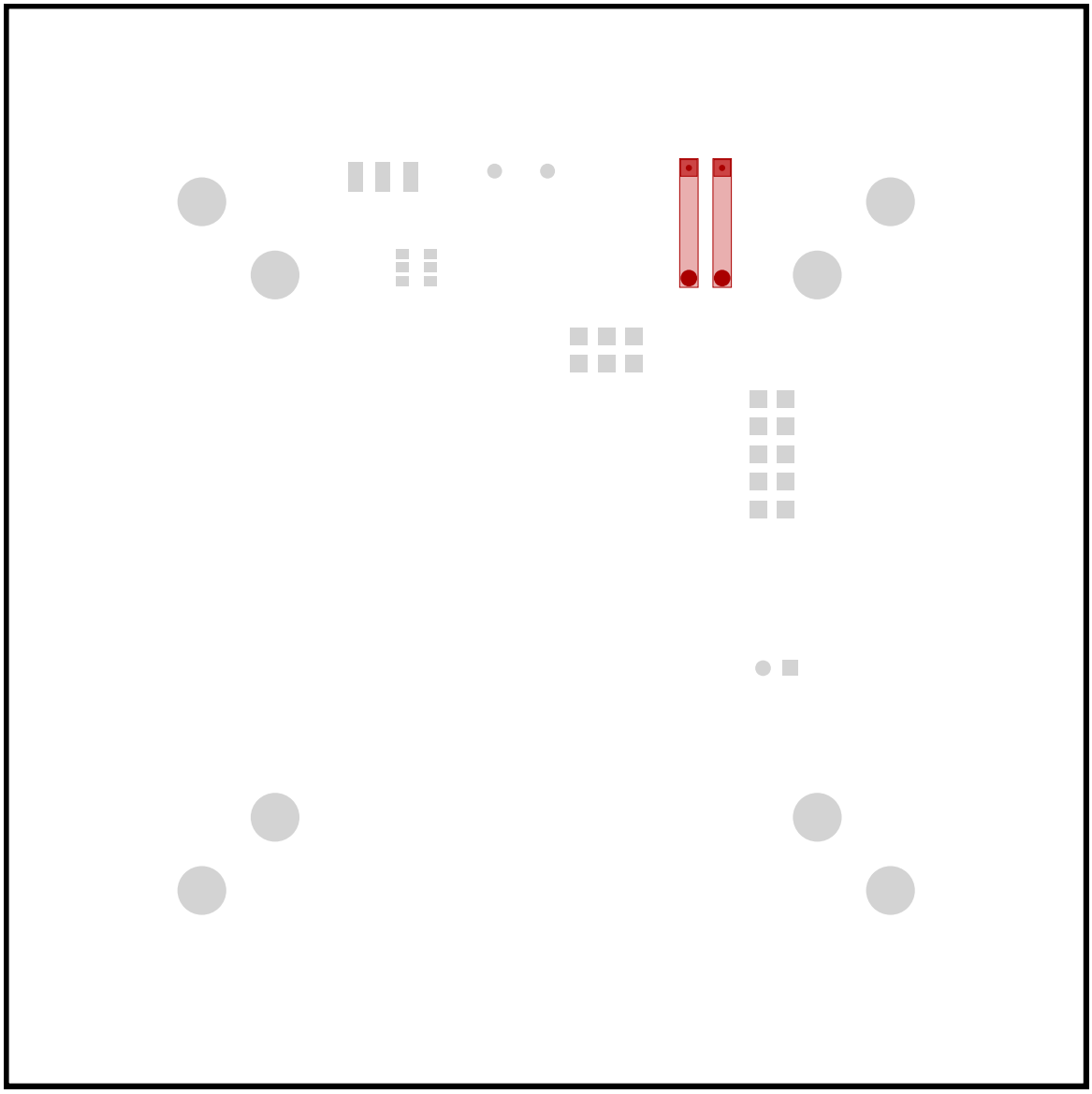
4.1 Oberseite

1x 10 - 100 uF, CP_Radial_Tantal_D8.0mm_P2.50mm



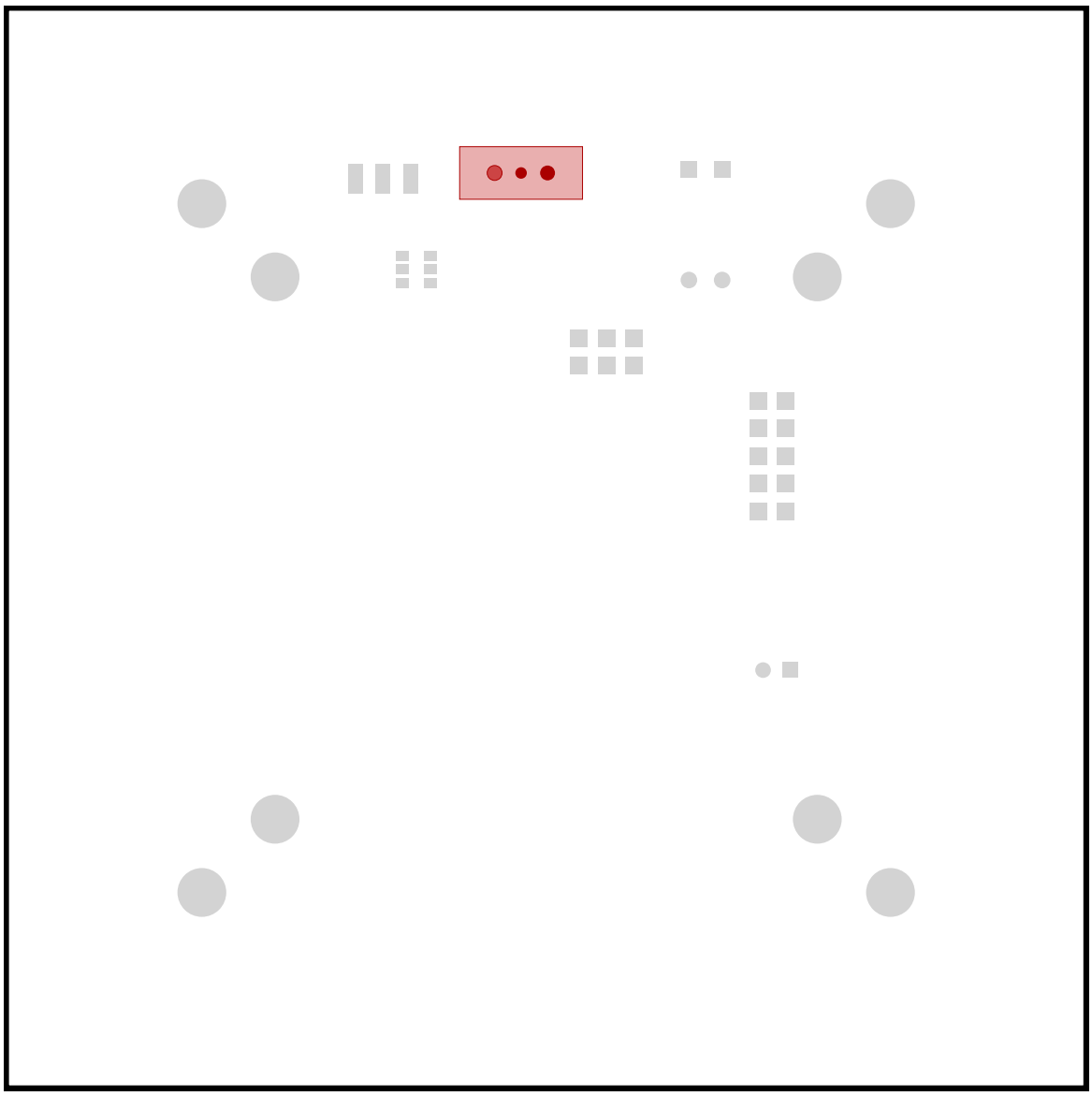
C23

2x D, Diode_throughole



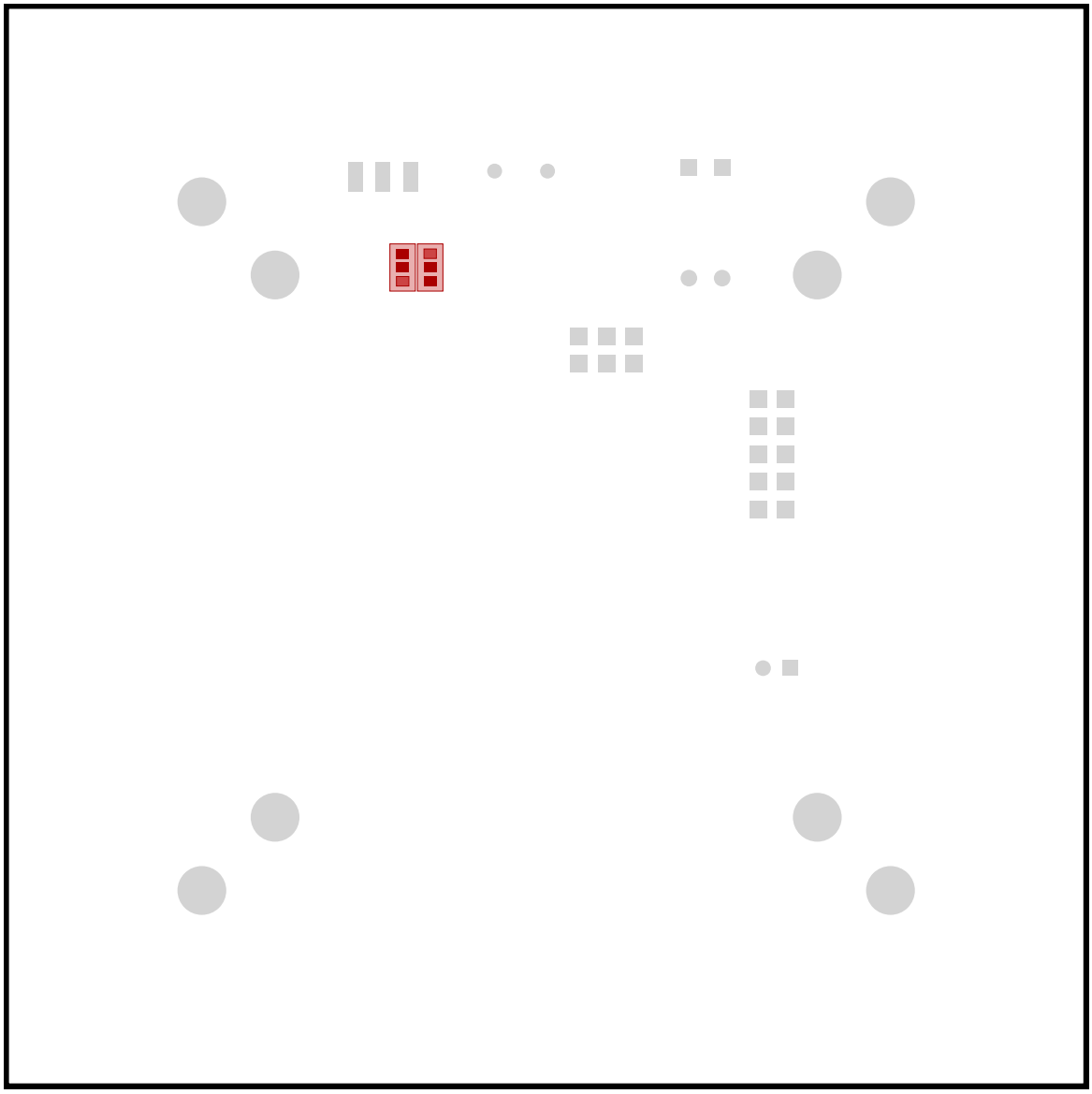
D34, D35

1x 16Mhz, QUARZ-HC49



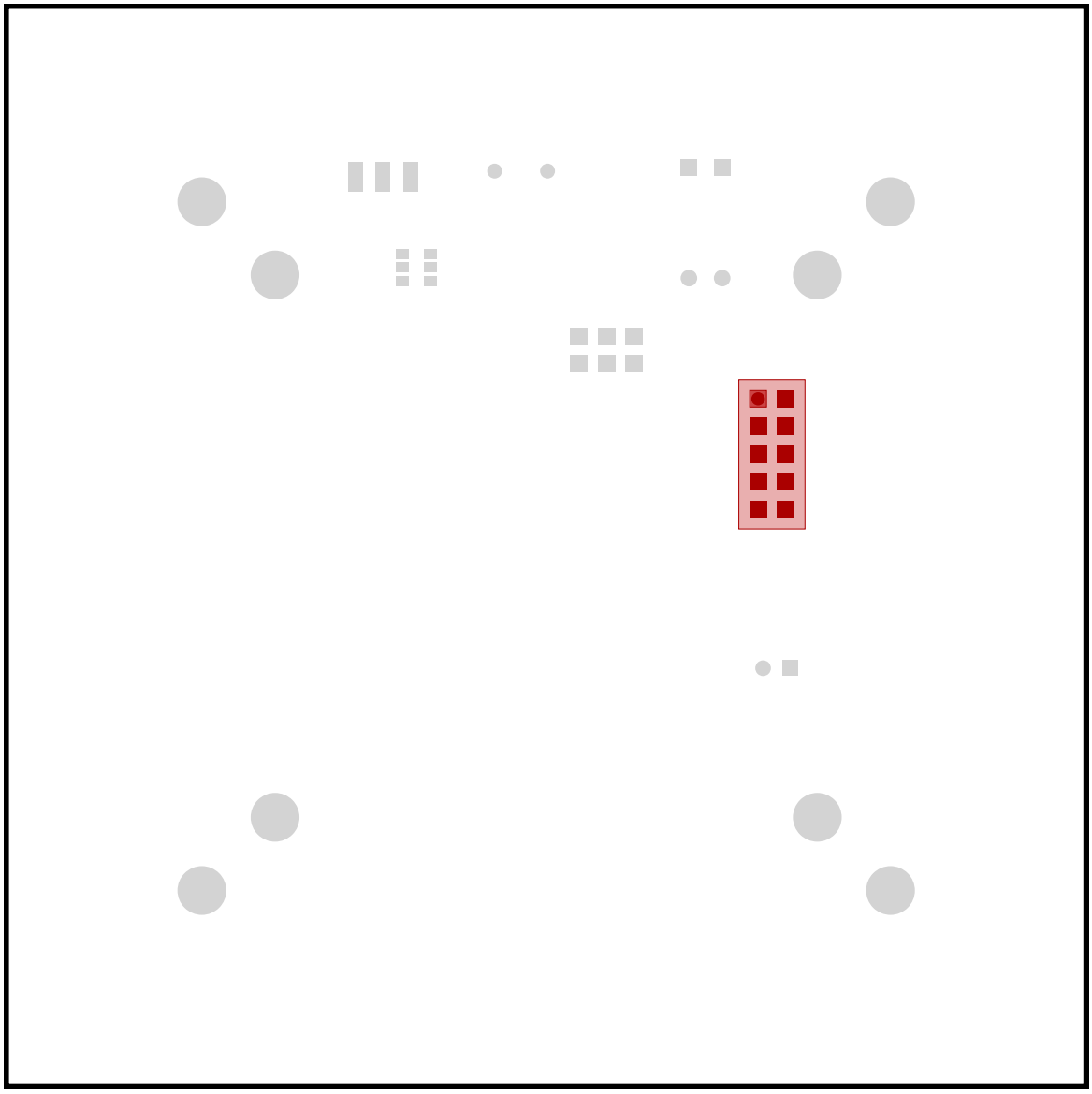
U3

2x Jumper_NC_Dual, GS3



JP1, JP2

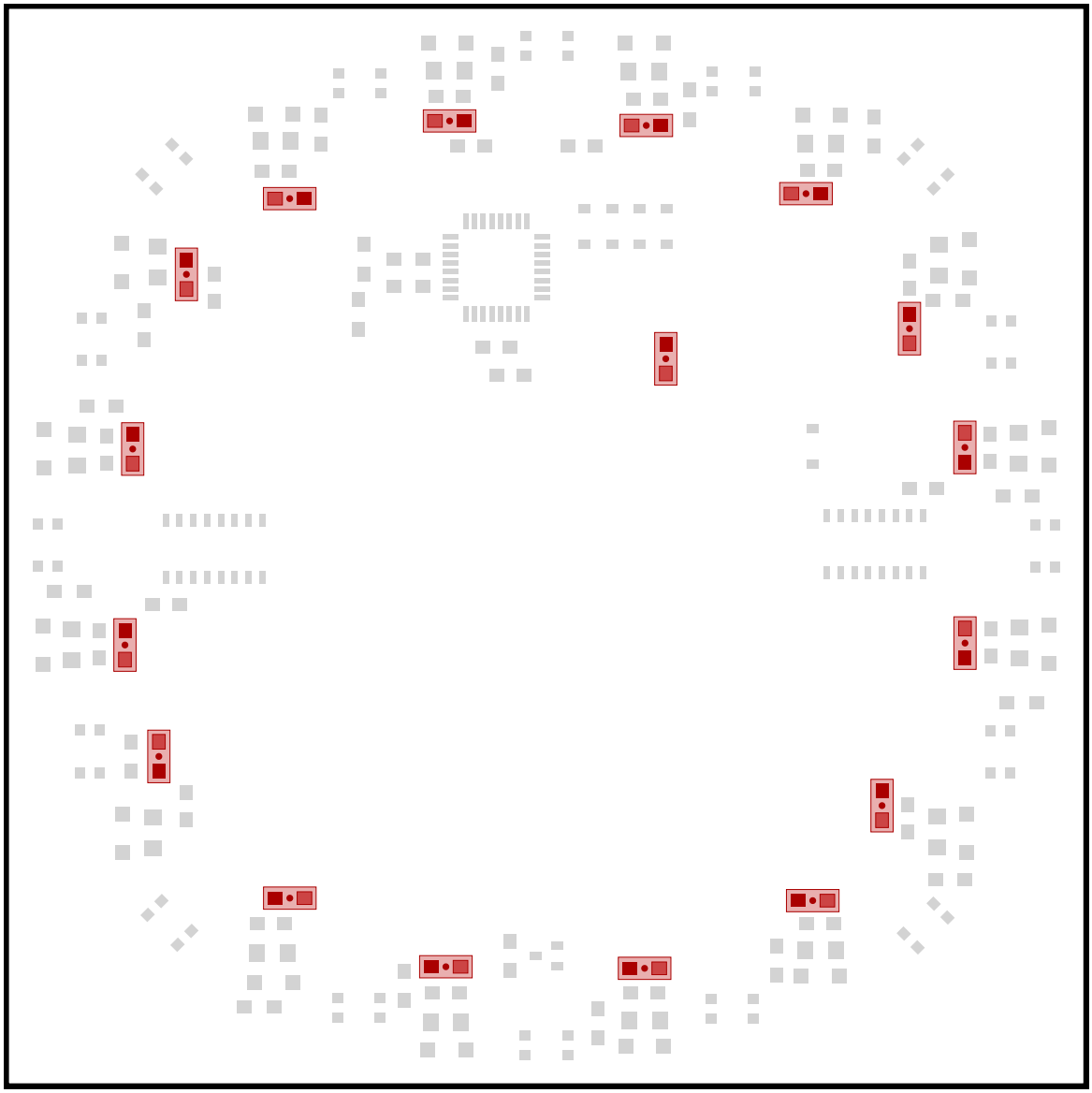
1x I2C, Wannenstecker_2x05



P1

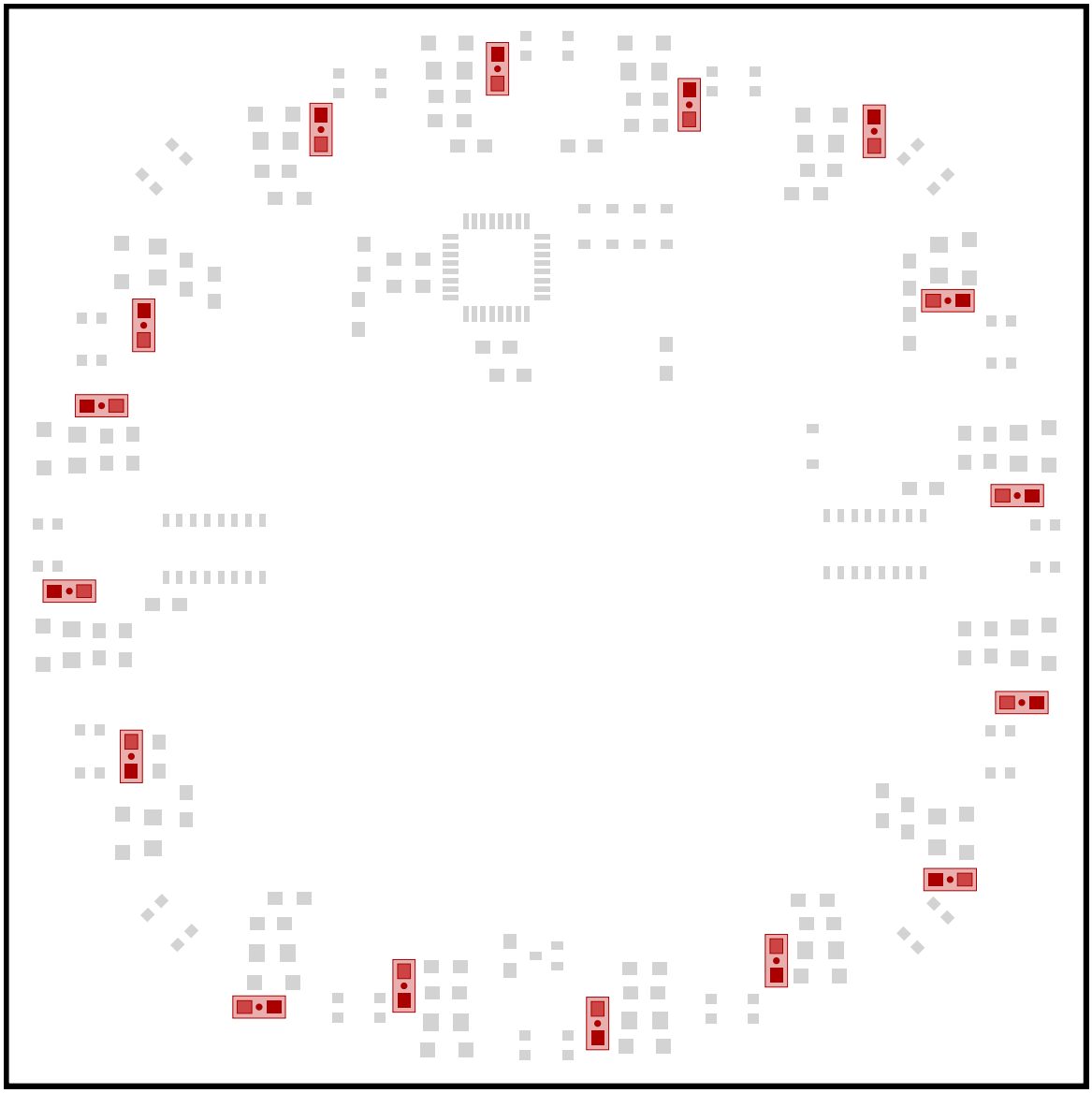
4.2 Unterseite

17x 10k, R_0805



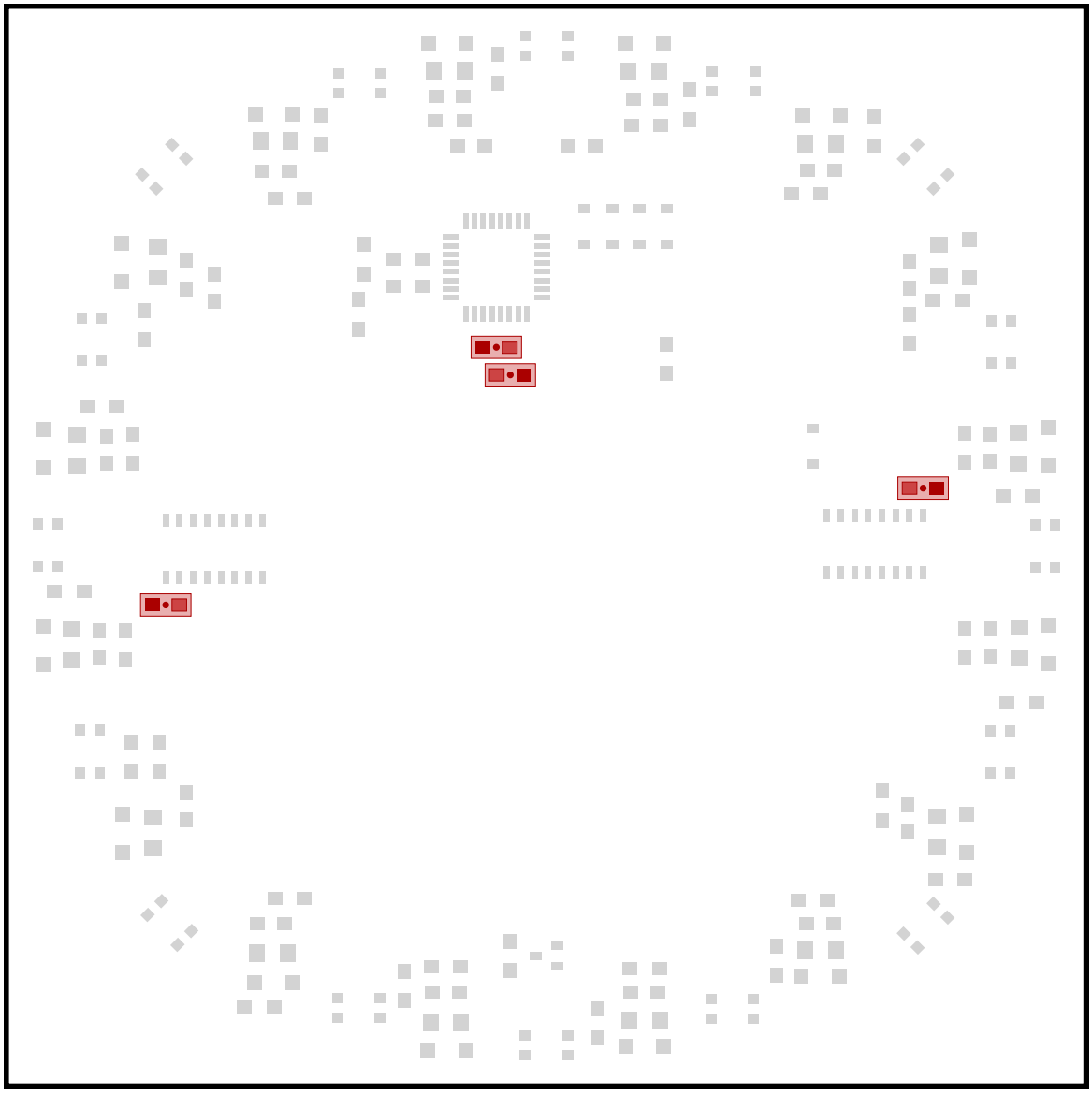
R5, R6, R7, R8, R13, R14, R15, R16, R21, R22, R23, R24, R29, R30, R31, R32, R67

16x 470, R_0805



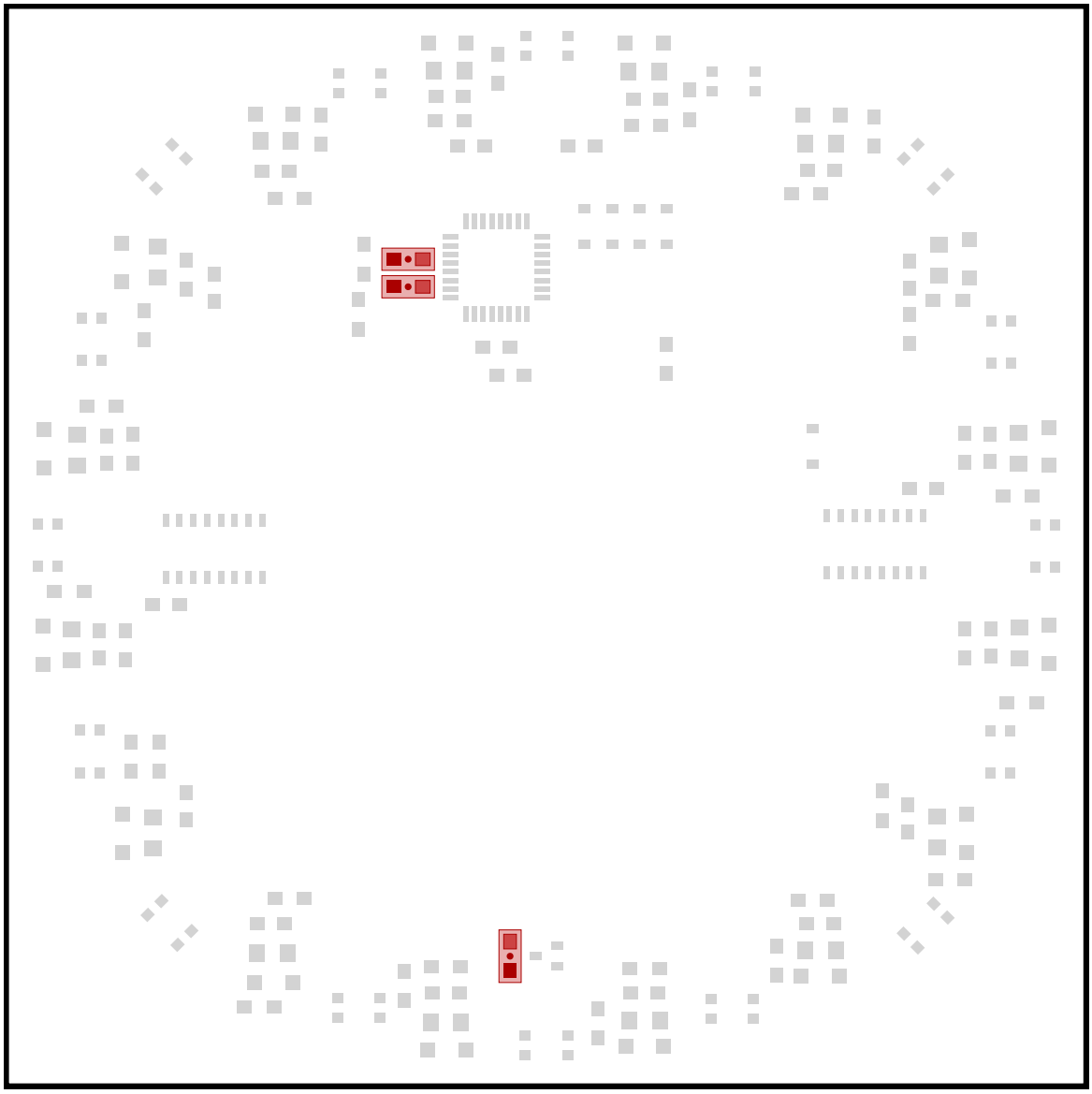
R1, R2, R3, R4, R9, R10, R11, R12, R17, R18, R19, R20, R25, R26, R27, R28

4x 100n, C_0805



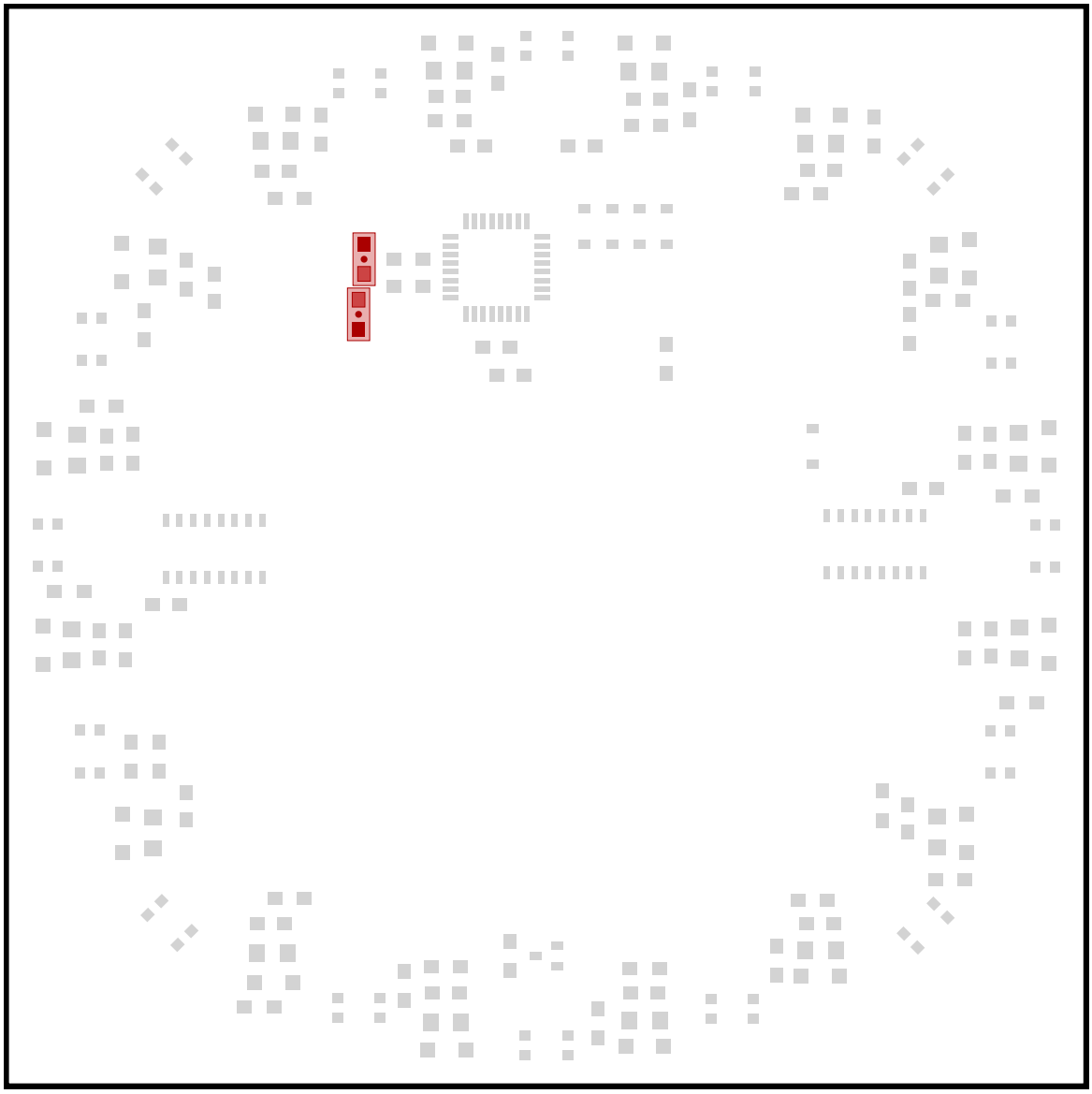
C11, C12, C13, C22

3x 1k, R_0805



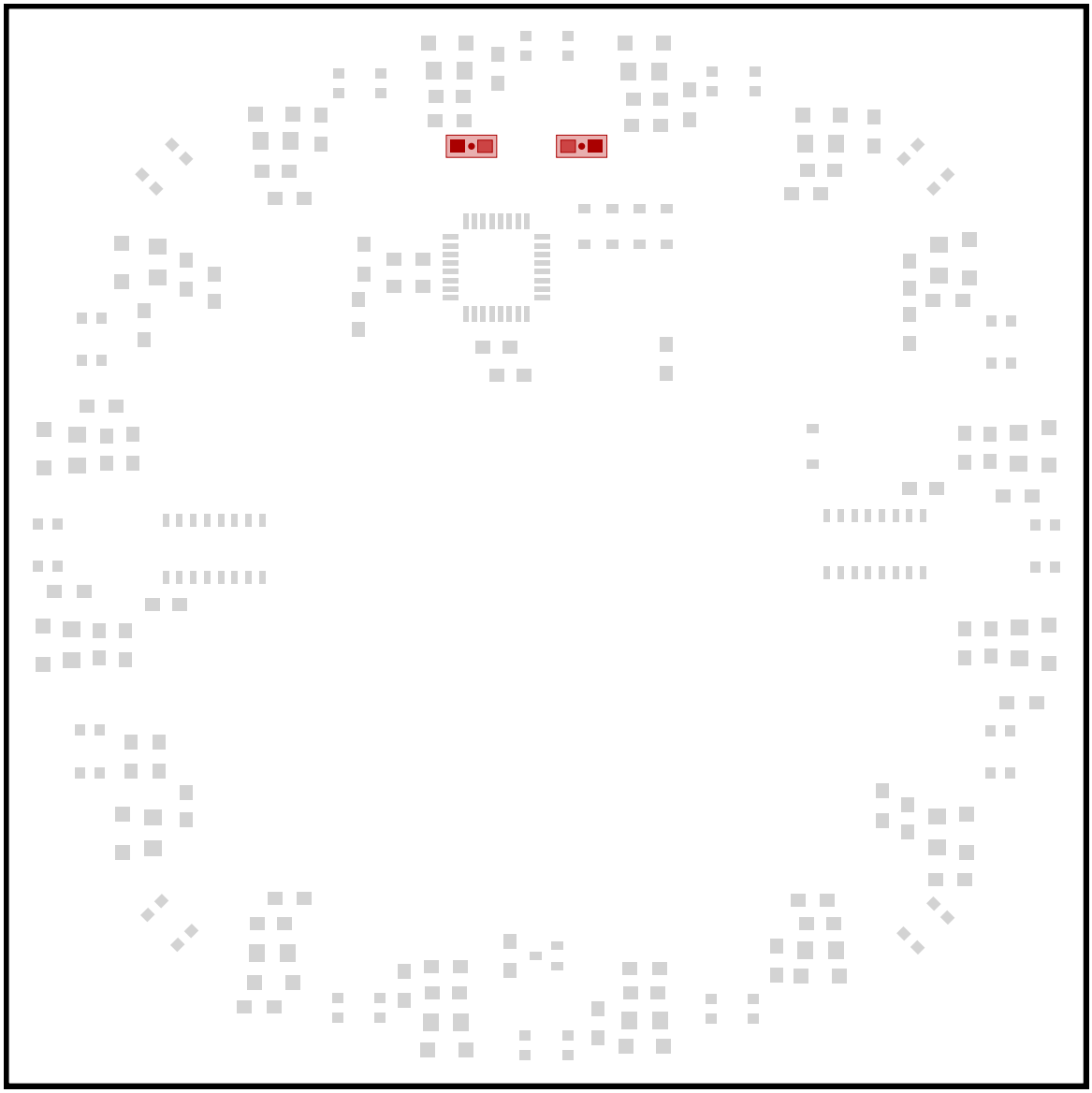
R33, R68, R69

2x 4,7k, R_0805



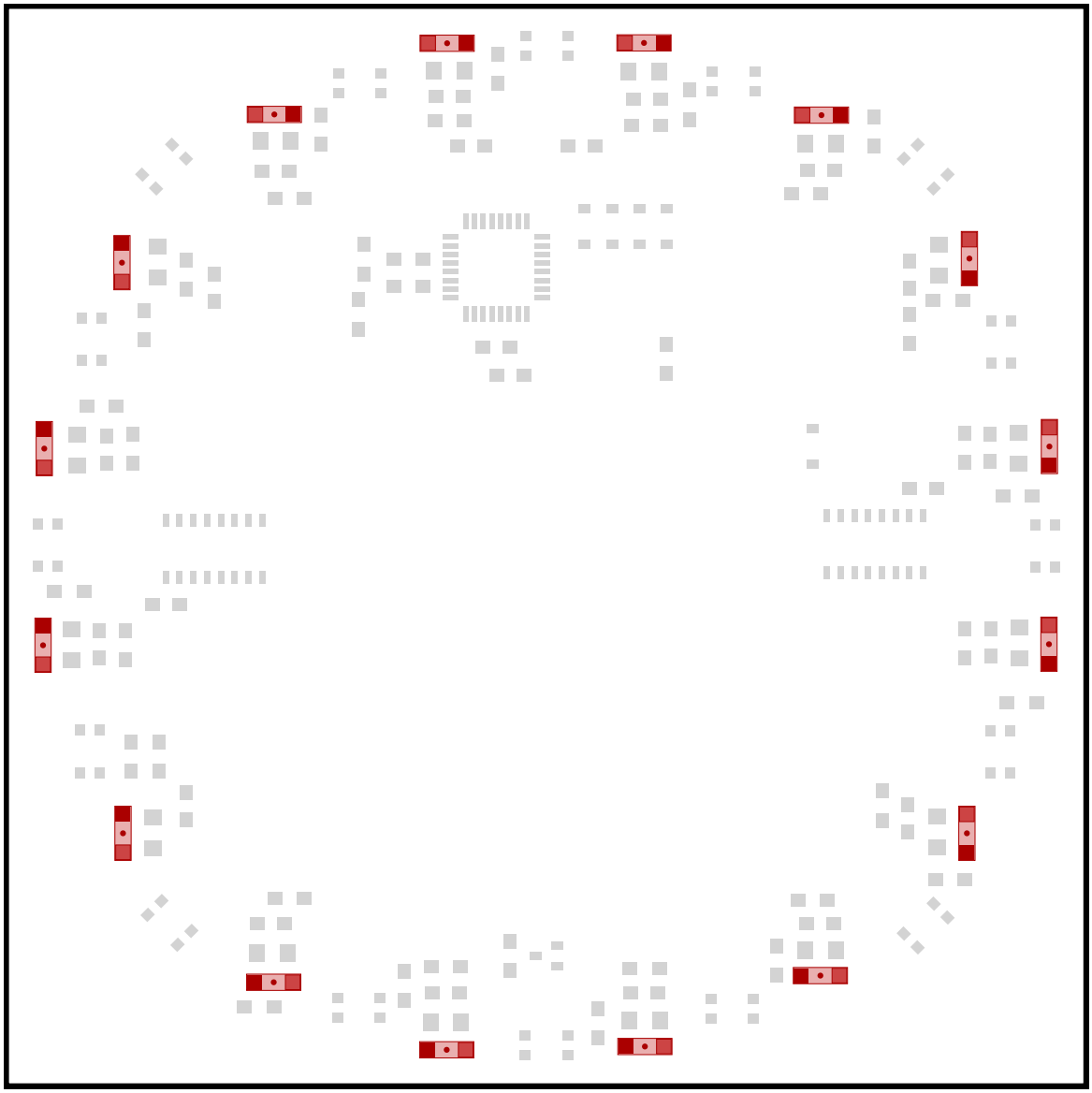
R65, R66

2x 22p, C_0805



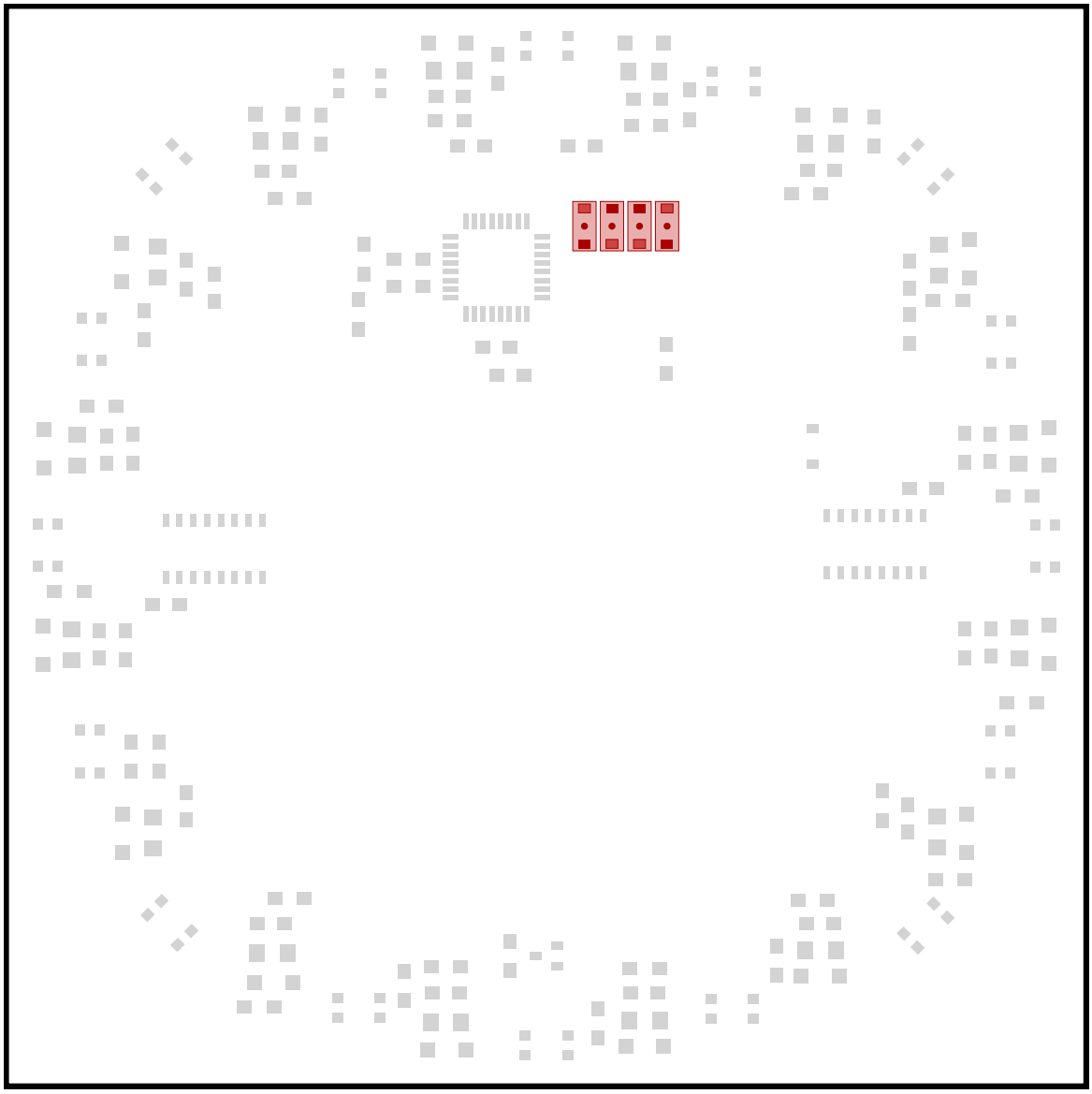
C9, C10

16x LED, 1206_LED



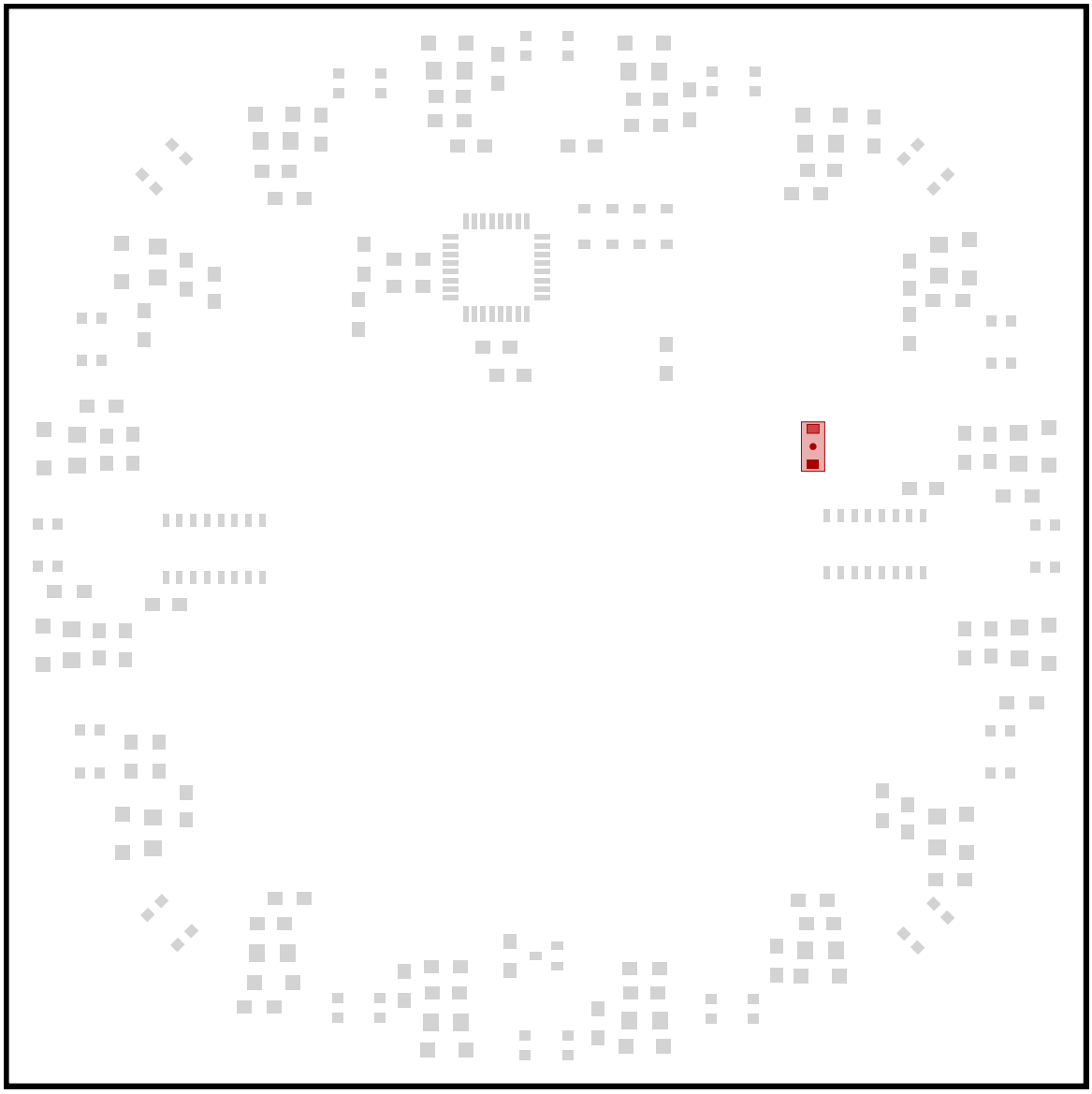
D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16

4x D_Schottky, Shotkey-SOD-123



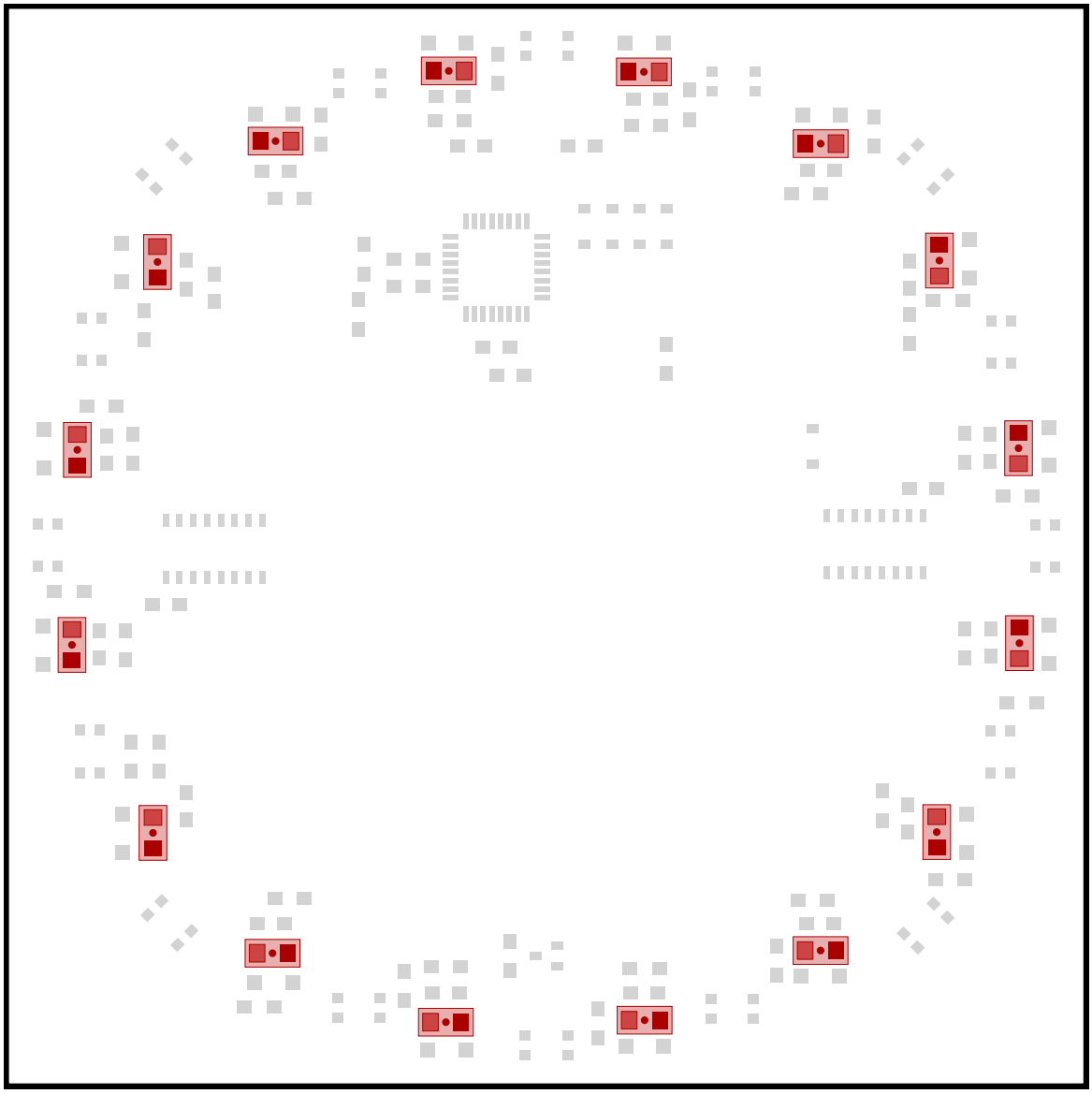
D40, D41, D42, D43

1x ZENER, SMD-shotkey



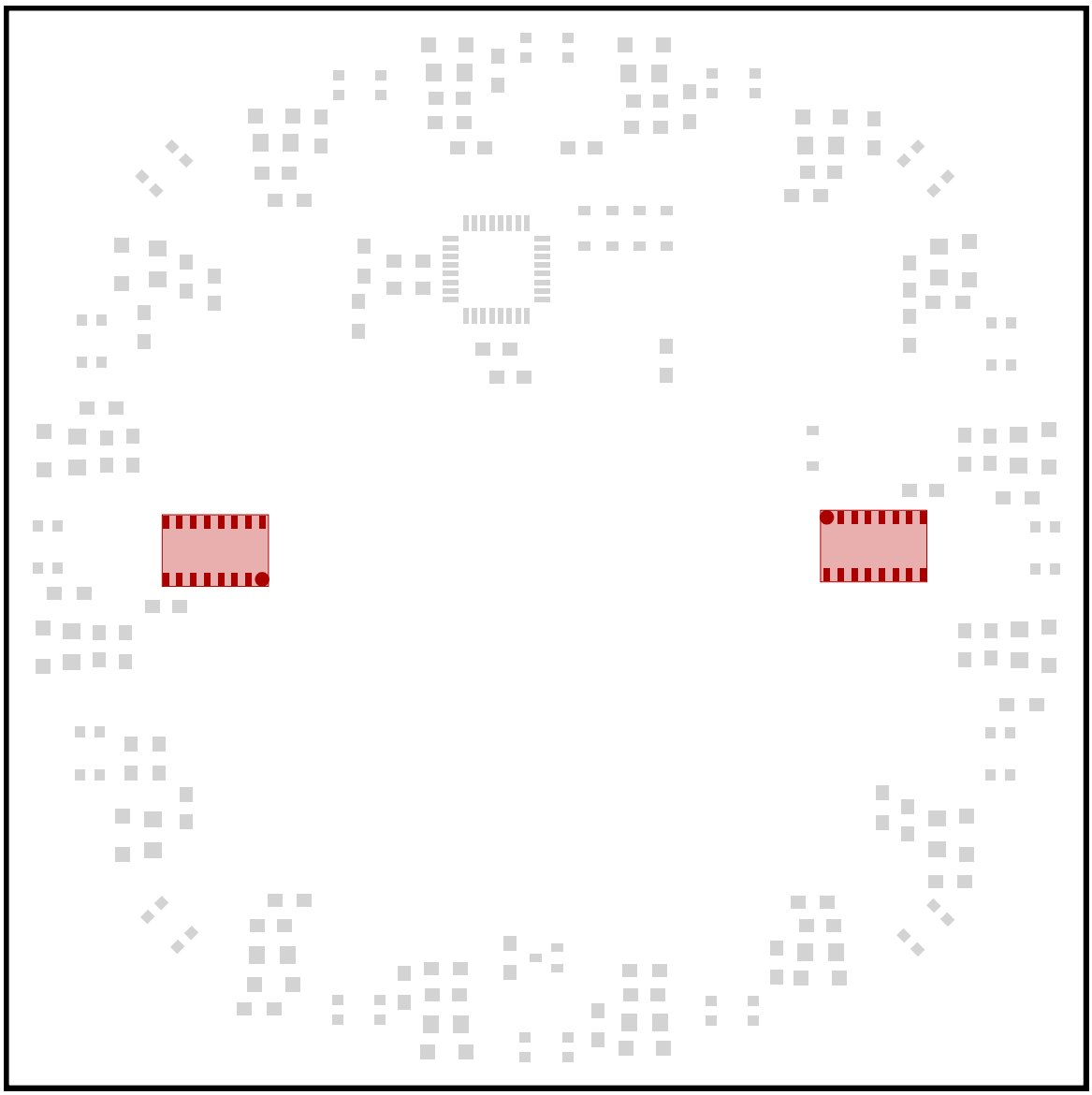
D44

16x SFH3710, SFH3710



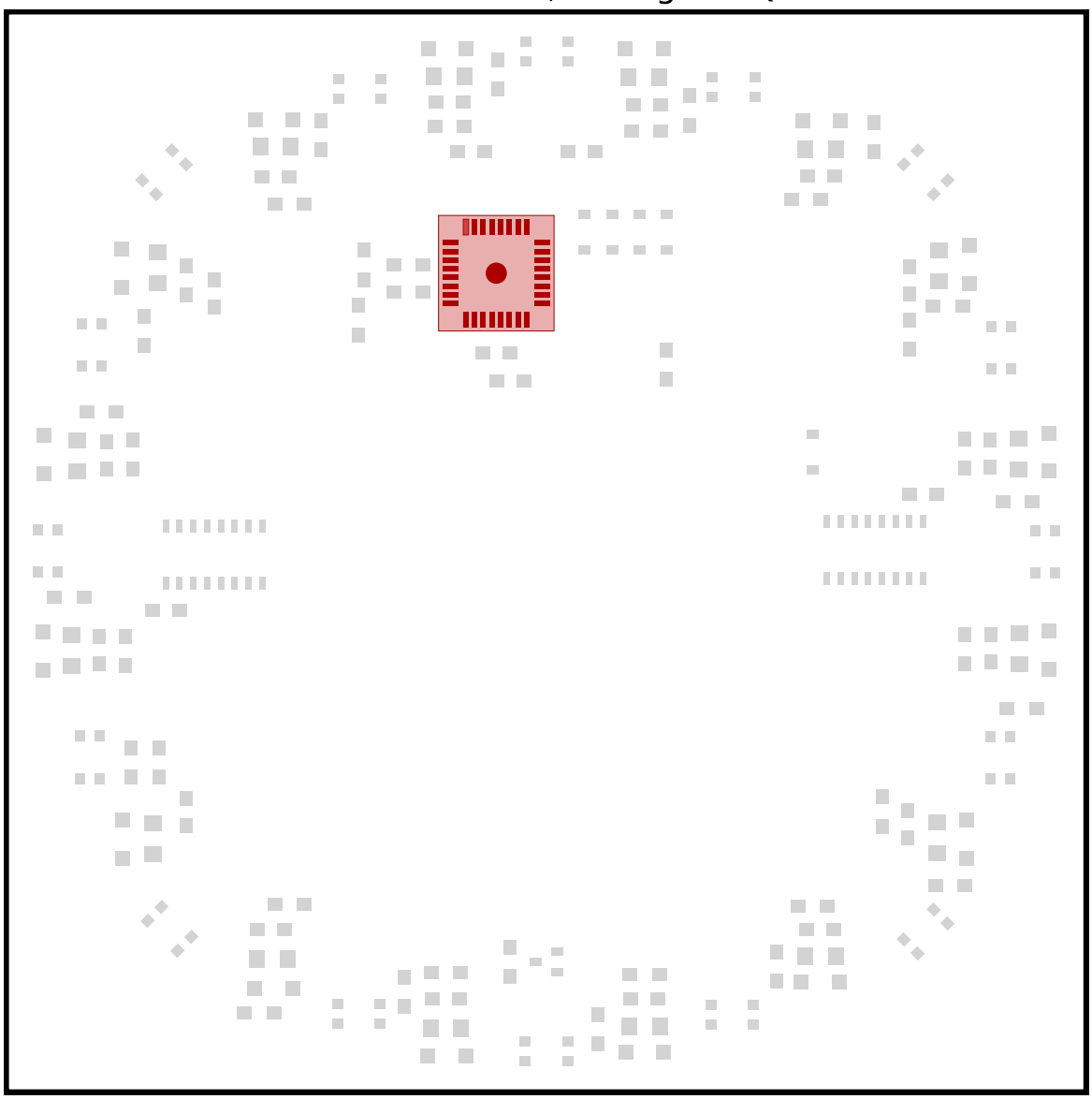
F1, F6, F7, F8, F13, F14, F15, F16, F21, F22, F23, F24, F29, F30, F31, F32

2x 74HC4051, 74HC4051-SO16



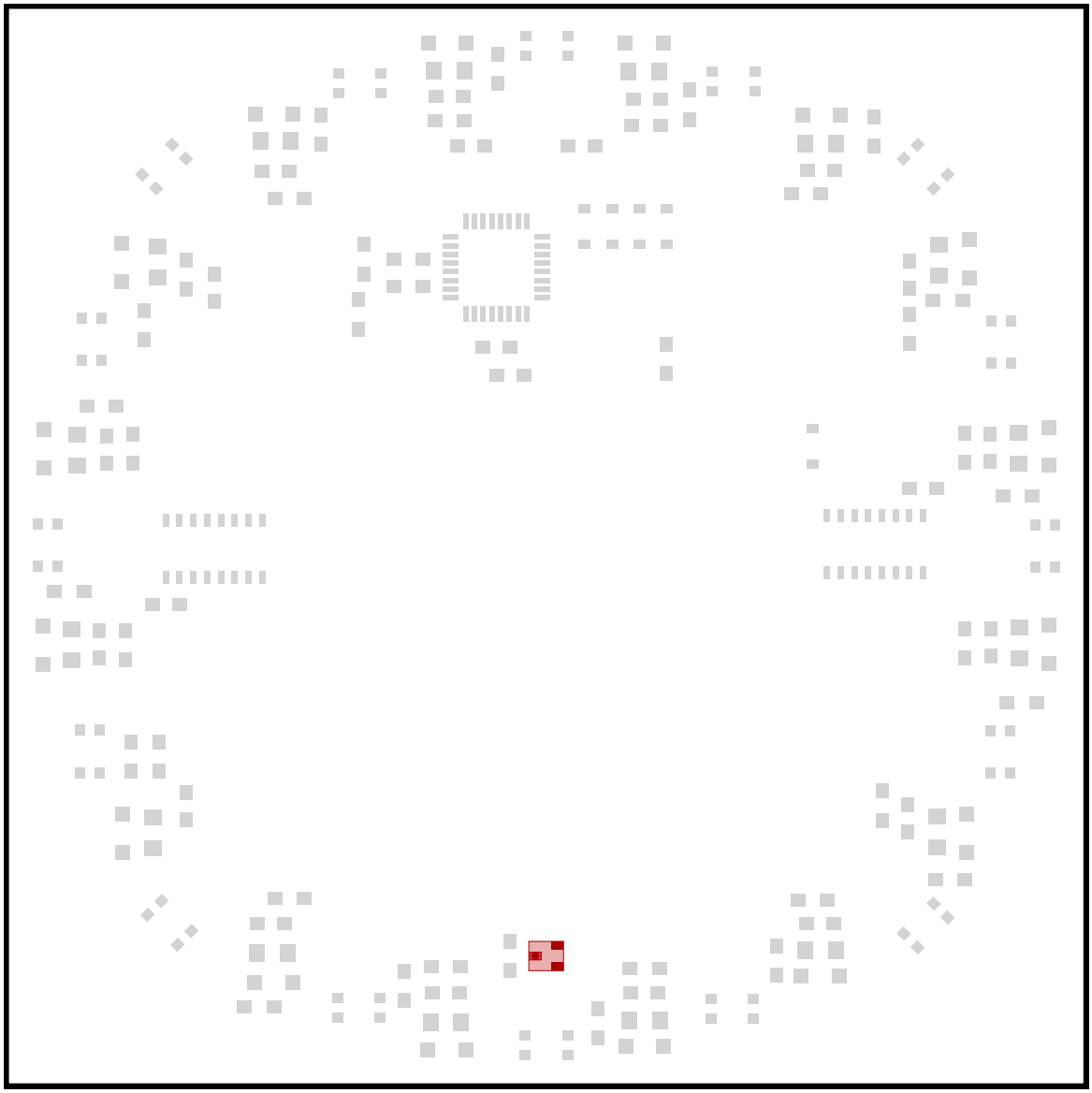
U1, U2

1x ATMEGA88-A, atmega8-TQFP



IC1

1x Q_NMOS_DGS, SOT23



Q1