Build manual PMR446 doorbell and intercom

M.T. Konstapel

2021-03-05

Abstract

The PMR446 doorbell and intercom is purpose build for my home situation: it is a doorbell which sends a signal over the PMR446 radio to the walkie-talkies we use when we are in our large garden. That way, we can hear the doorbell ring and we even can talk back to the intercom unit, which is part of the design. This version is a redesign of the original one, which was build around five years ago.

1 An overview

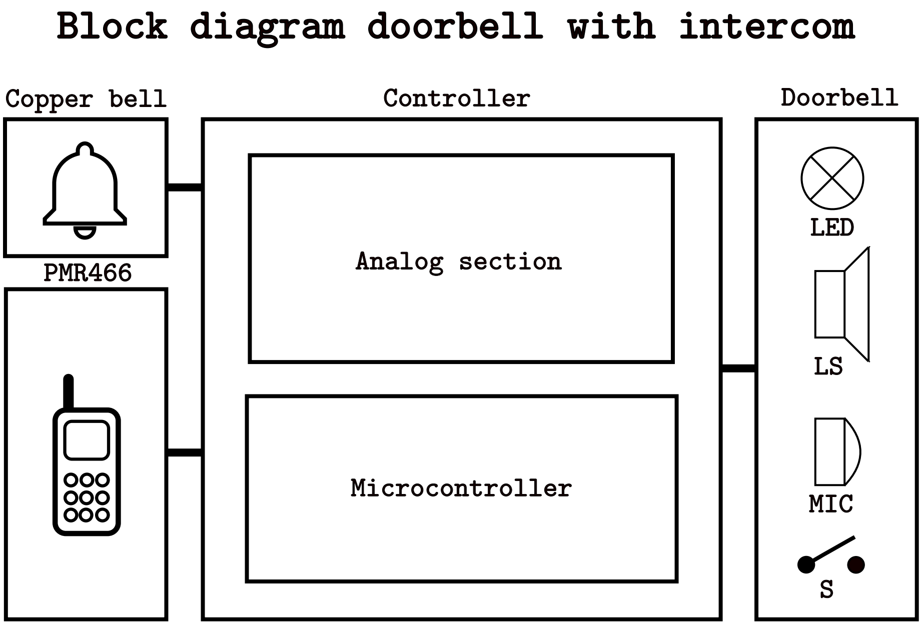

Figure 1: block diagram of the doorbell

Figure 1 shows the block diagram of the doorbell. On the right you can see the intercom module, which is located by the door. It consists of a button, a microphone, a loud speaker and an indicator-LED. When the button is pushed, it sends a signal to the microcontroller. The controller rings the copper bell and synthesise a signal. This signal is processed by the analog section and goes back to the loudspeaker of the intercom, indicating that the bell is indeed pressed. And it is also being transmitted by the radio. After ringing the bell for four times, the intercom can be used as an ordinary walky-talky. So it is possible to have a conversation without coming to the door. When the intercom is not used for more than two minutes, it switches off and waits for the next visitor.

In the old design, I used a library to synthesize the bell signal. It generated a simple 1 kHz pulsating signal. Very effective, but not very sophisticated.

2 A new ring tone

For the new design, I am inspired by the old fashioned telephone dial tone. As shown in figure 2, it is made by mixing two single tones both around 400 Hertz, but about 100 Hertz spaced appart. In the example a tone of 440 Hertz and a tone of 350 Hertz. The difference, which in this case is 90 Hertz, is called the beat. This interfering signal is audible as a periodic variation in volume. It sounds very smooth and rich compared to a single tone. This dial tone is periodically switch on and off: 0.4 seconds on, 0.4 seconds off, 0.4 seconds on and 1.6 seconds off.

The dial tone is generated by an interupt service routine, triggered by the real time clock. The interrupt is triggered every 375.3 microseconds. The signal of 1332Hertz is half of that and is generated by incrementing a counter at every interrupt. Bit 0 of this counter now toggles with a frequency of 1332 Hertz. Bit 2 of this counter has a frequency of 333 Hz, Bit 10 has a frequency of 1.3 Hertz, and bit 12 has a frequency of 0.325 Hertz. Generating a signal of 444 Hertz is more comlicated, because it is a division by three, but it only needs a couple of lines of code. The source code, as well as all the design files, are available on my website. The AND-gates switch the signals on and off, creating the distingt telephone dial tone.

Both signals come out of the microcontroller on seperate pins. Two resisters combine these signals. A low pass filter smooths the signal, as it can sound a little harsh.

Dial tone generator

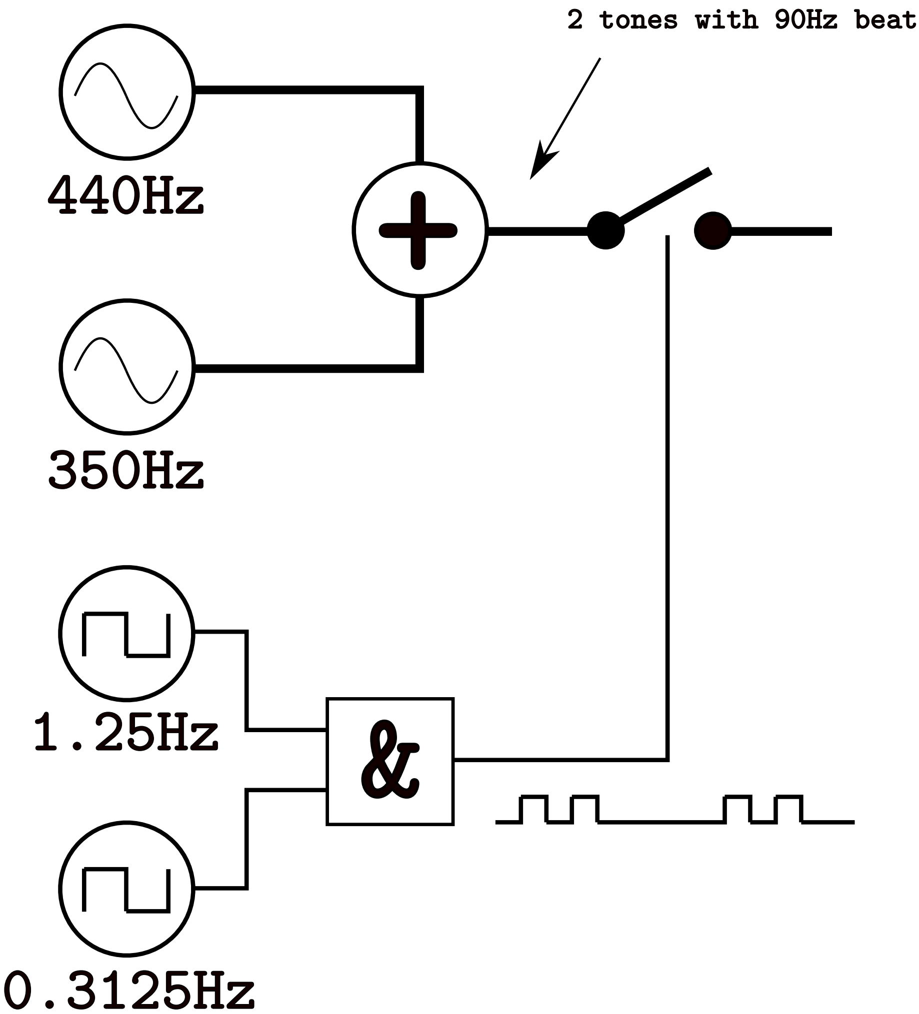

Figure 2: The new ring tone

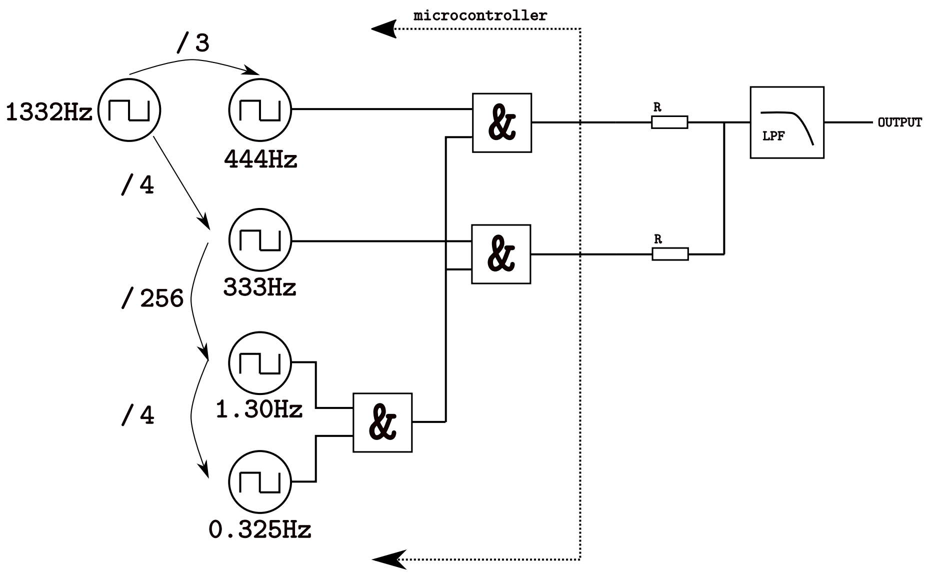

Figure 3: The generation of the dial tone

3 Design

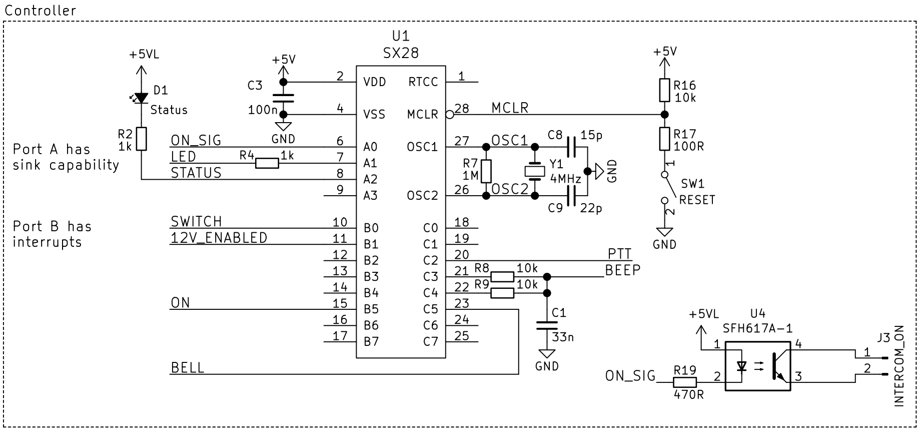

Figure 4: microcontroller

The design is straight forward: figure 4 shows the SX28, which is the heart of the operation. It is clocked with a 4 MHz crystal. I used a 4.433619MHz crystal, as I have a lot of these laying around: the timing in the program is based on this frequency.

The optocoupler is an optional feature: it can be used to interface to an other device, an IoT for example. I do not use this yet.

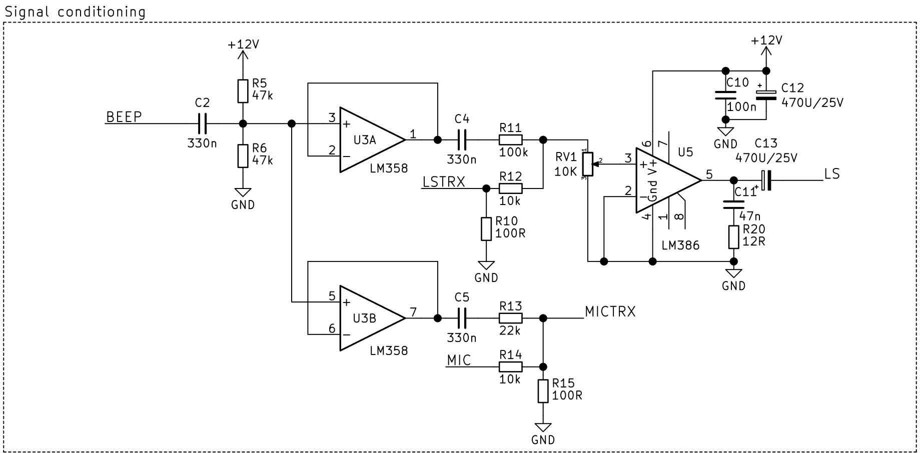

Figure 5: analog section

Figure 5 shows the analog section. Here all the analog signals are conditioned and mixed. A small amplifier amplifies the signal going to the loudspeaker in the intercom.

The copper bell is powered by a MOSFET (figure 6) and for the PMR446, you can choose between a separate PTT signal or a combined PTT/microphone signal by setting the corresponding jumper on JP2 (see figure 7).

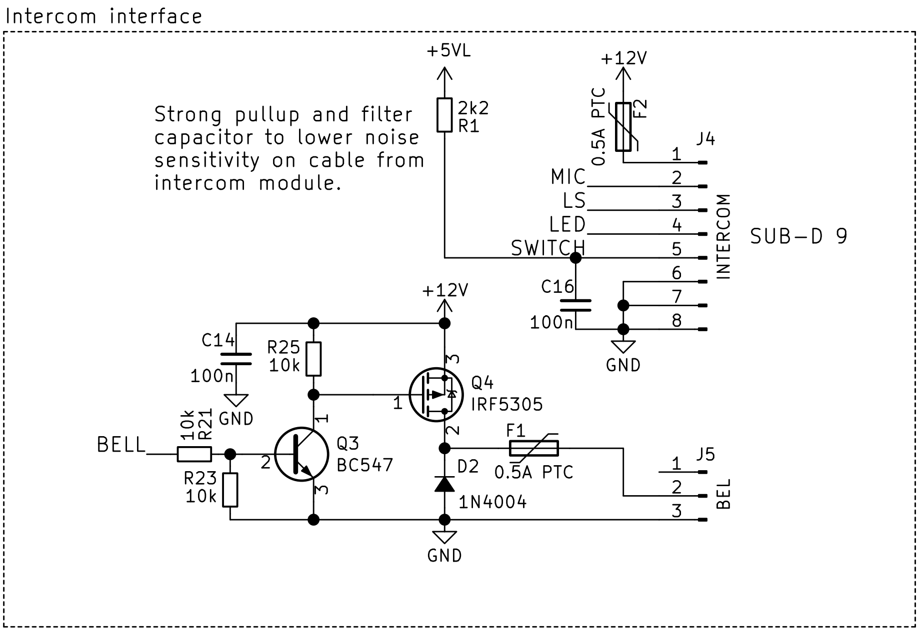

Figure 6: analog section

Everything exept the microcontroller is powered by a 12 Volt boost converter (figure 8). This converter is switched off when the processor waits for a button press, resulting in a current drawn of only several microamps.

The output of Q1 is low when the DC/DC-converter outputs 12 Volts. The microcontroller can use this a check. This is not implemented in the current firmware.

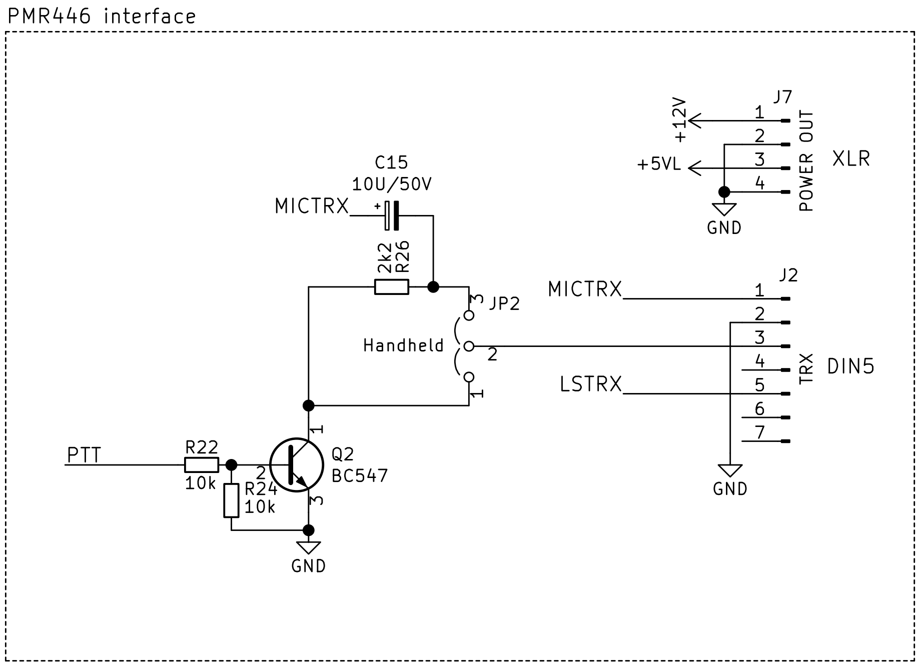

Figure 7: analog section

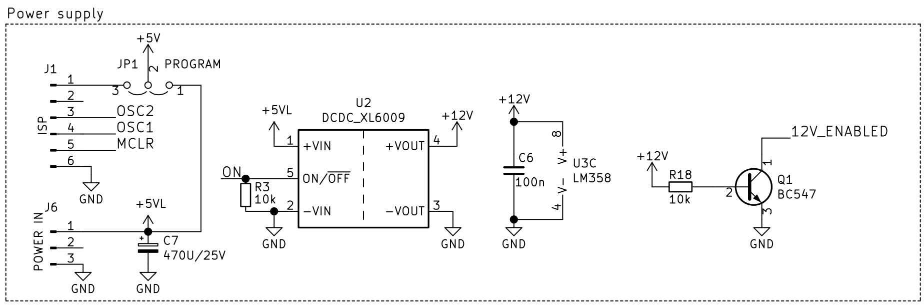

Figure 8: analog section

4 Practical notes

4.1 Build the pcb

Building the printed circuit board is straight forward. The design files are made with KiCad 5.1.2 and scaled PDF files of the printed circuit board are available. The pcb has two layers and I etched it myself. Therefore, there are no plated through vias. All ground connections should be soldered on both sides of the pcb, as well as some other connections.

A Full schematic

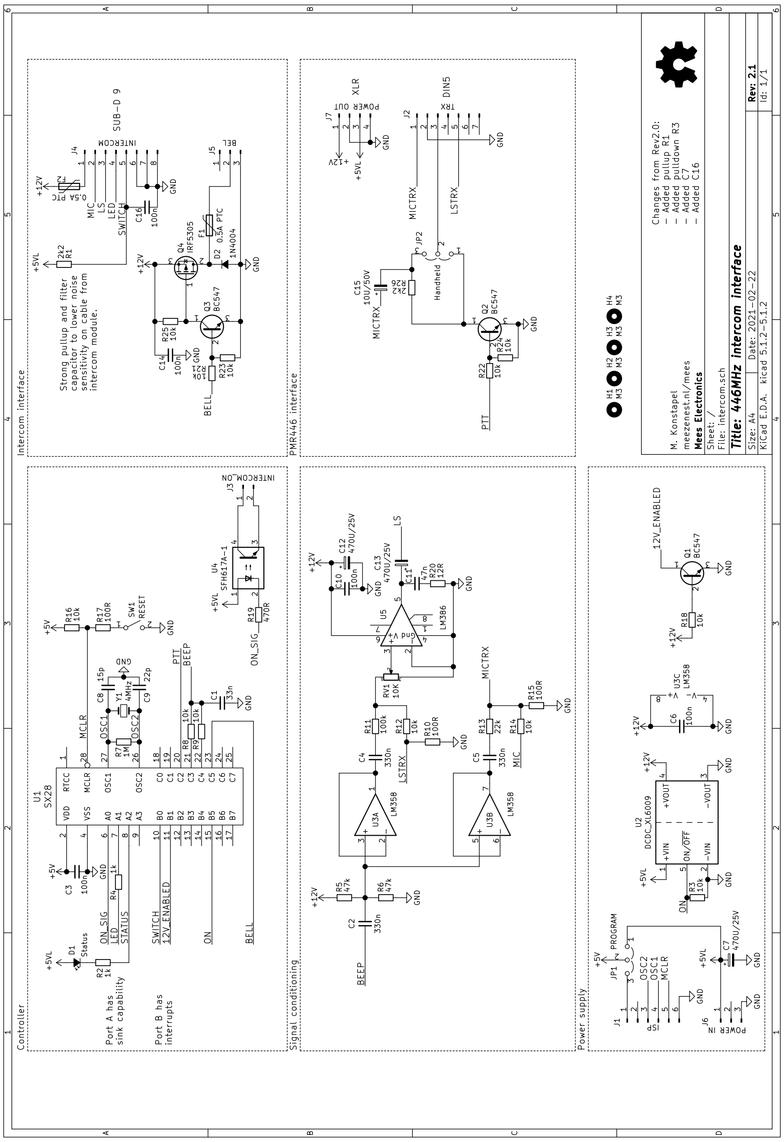

Figure 9: full schematic of wireless doorbell

B Component placement

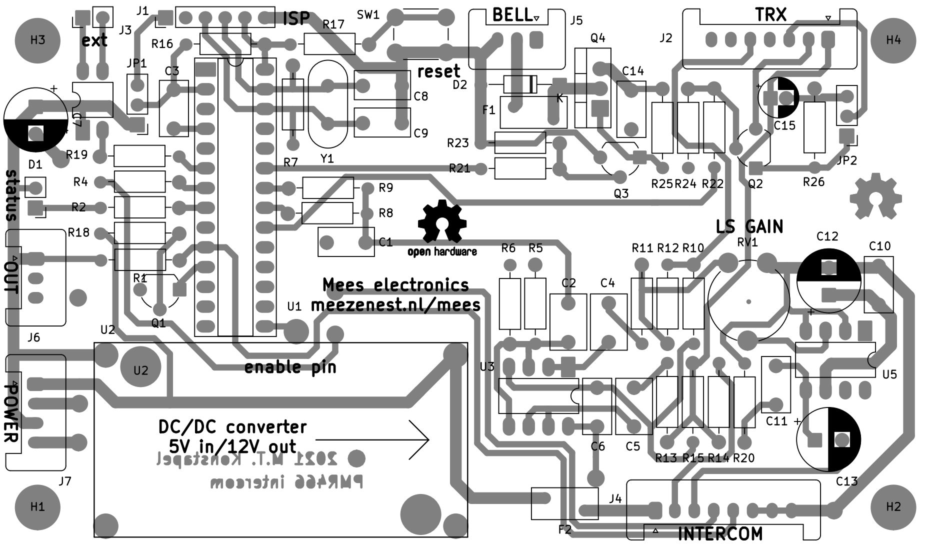

Figure 10: component placement

C Bill of material

#	References	Description	Value	Ordering #	Manufacturer	Conrad #
5	C3 C6 C10 C14 C16	Capacitor	100n	100n/50V	QPL	1584724
1	C15	Electrolytic capacitor	10U/50V	EEUFR1H100	Panasonic	1675308
1	C8	Capacitor	15p	15p/100V	QPL	1589403
1	C9	Capacitor	22p	22p/100V	QPL	1589403
3	C2 C4 C5	Capacitor	330n	330n/50V	QPL	1578738
1	C1	Capacitor	33n	33n/100V	QPL	1578715
3	C7 C12 C13	Electrolytic capacitor	470U/25V	EEUFR1E471Y	Panasonic	792079
1	C11	Capacitor	47n	47n/100V	QPL	1589452
1	D2	Diode	1N4004	1N4004	QPL	162248
1	D1	LED, red	Status	Header 1x2 2.54mm	QPL	-
2	F1 F2	Resettable fuse	0.5A PTC	FRX050-90F	ESKA	524774
2	J5 J6	3 pole Stocko	BEL	3 pole	Stocko	-
1	J4	8 pole Stocko	INTERCOM	8 pole	Stocko	-
1	J3	2 pole header 2.54mm pitch	INTERCOM_ON	Header 1x2 2.54mm	QPL	-
1	J1	6 pole header 2.54mm pitch	ISP	Header 1x6 2.54mm	QPL	-
1	J7	4 pole Stocko	POWER OUT	4 pole	Stocko	
1	J2	7 pole Stocko	TRX	7 pole	Stocko	-
1	JP2	3 pole header 2.54mm pitch	Handheld	Header 1x3 2.54mm	QPL	-
1	JP1	6 pole header 2.54mm pitch	PROGRAM	Header 1x6 2.54mm	QPL	-
3	Q1 Q2 Q3	Transistor NPN	BC547	BC547	QPL	140539
1	Q4	P-channel HEXFET	IRF5305	IRF5305	Infineon	162408
3	R10 R15 R17	Resistor	100R	100R/1%/0.6W	QPL	1585430
1	R11	Resistor	100k	100k/1%/0.6W	QPL	1556968
12	R3 R8 R9 R12 R14 R16 R18 R21 R22 R23 R24 R25	Resistor	10k	10k/1%/0.6W	QPL	1556969
1	R20	Resistor	12R	12R/1%/0.6W	QPL	1584928
1	R7	Resistor	1M	1M/1%/0.6W	QPL	1556889
2	R2 R4	Resistor	1k	1k/1%/0.6W	QPL	1557083
1	R13	Resistor	22k	22k/1%/0.6W	QPL	1557290
2	R1 R26	Resistor	2k2	2k2/1%/0.6W	QPL	1557513
1	R19	Resistor	470R	470R/1%/0.6W	QPL	1557322
2	R5 R6	Resistor	47k	47k/1%/0.6W	QPL	1557224
1	RV1	Potentiometer	10K	PT 10 LV 10K	Piher	430862
1	SW1	Switch	RESET	Generic switch 4 pin	QPL	701749
1	U2	DC/DC converter	DCDC_XL6009	XL6009 DC/DC	QPL	-
1	U3	Dual opamp	LM358	LM358	QPL	-
1	U5	Audio amplifier	LM386	LM386	QPL	-
1	U4	Optocoupler	SFH617A-1	SFH617A-1	Vishay	153819
1	U1	Microcontroler	SX28	SX28	Scenix	-
1	Y1	Crystal	4MHz	4MHz	QPL	182087

Figure 11: bill of material

D Outside bell module

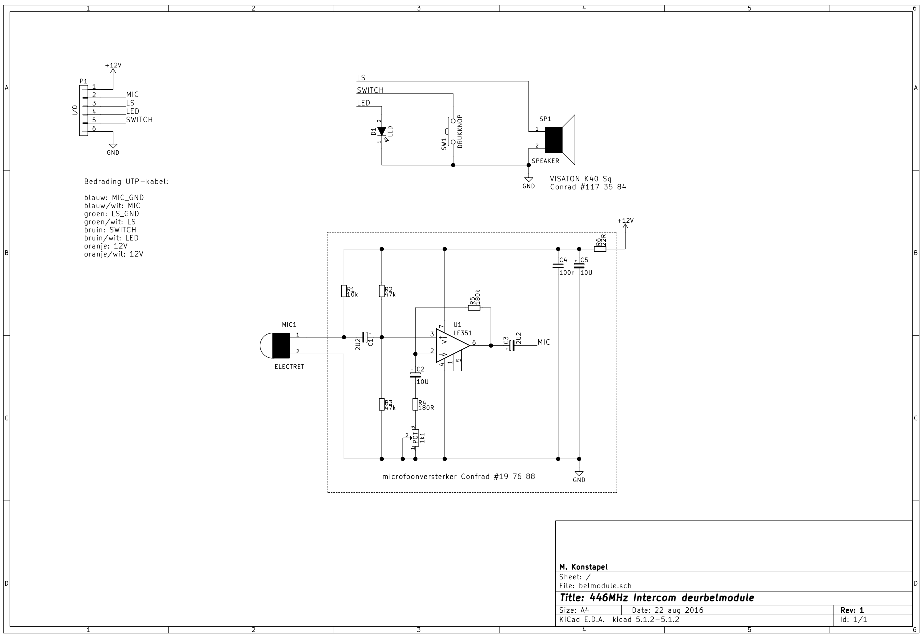

Figure 12: schematic of outside intercom module

E Open source hardware

All the design files are available on my website: https://www.meezenest.nl/mees