



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STAVEBNÍ**

FACULTY OF CIVIL ENGINEERING

**ÚSTAV POZEMNÍCH STAVEB**

INSTITUTE OF BUILDING STRUCTURES

**NÁVRH AKUSTICKY POHLTIVÝCH KONSTRUKCÍ  
S POUŽITÍM DĚROVANÝCH DESEK**

DESIGN OF ACOUSTIC ABSORBERS USING PERFORATED BOARDS

**PŘÍLOHY**

APPENDICES

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. David Jun**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Lubor Kalousek, Ph.D.**

**BRNO 2021**



# Obsah

<b>Přílohy</b>	<b>115</b>
<b>A Výběr skriptů souvisejících s měřením</b>	<b>119</b>
A.1 Měření měrného odporu proti proudění vzduchu . . . . .	119
A.1.1 Definování hlavních funkcí pro práci s měřicí buňkou . . . . .	119
A.1.2 Skript používaný pro vlastní měření . . . . .	122
A.2 Měření doby dozvuku . . . . .	123
A.2.1 Výňatek z modulu acoustics.room . . . . .	123
A.2.2 Skript vlastního měření doby dozvuku . . . . .	126
<b>B Výběr z fotodokumentace</b>	<b>129</b>
B.1 Měření měrného odporu proti proudění vzduchu . . . . .	129
B.1.1 Měřicí buňka . . . . .	129
B.1.2 Foto z měření . . . . .	132
B.2 Měření činitele zvukové pohltivosti . . . . .	133
B.2.1 Příprava dozvukové místnosti . . . . .	133
B.2.2 Výroba desek . . . . .	137
B.2.3 Z průběhu měření . . . . .	140





# Příloha A

## Výběr skriptů souvisejících s měřením

Tato kapitola ukazuje některé základní skripty, které byly použity v rámci měření pro účely této práce. Z důvodu rozsahu nebylo možné a ani vhodné umístit veškerý kód, který vzniknul a který byl použit, nicméně je dobré si všimnout v rámci hlavičky každého skriptu importu množství modulů vytvořených pro účely této práce. Další skripty byly součástí dalšího processingu dat nebo jejich grafického ztvárnění. Zajímavostí pak může být, že celkový počet řádků kódu vzniklého pro účely této práce výrazně přesáhnul 4000, což odpovídá více než 130 stranám textu o 30 řádcích.

Zde uvedené skripty byly vybrány z důvodu ilustrace celkového přístupu k softwarovému řešení a k ilustraci jeho komplexnosti.

### A.1 Měření měrného odporu proti proudění vzduchu

#### A.1.1 Definování hlavních funkcí pro práci s měřicí buňkou

```
1  """
2  Fr_measurement
3  =====
4
5  Tento modul obsahuje stezejni funkce umoznujici mereni
6  merneho odporu proti proudeni vzduchu dle CSN EN ISO 9053-1
7  """
8
9  import pandas as pd
10 import numpy as np
11 import time as t
12 import os
13 import RPi.GPIO as GPIO
14
15 # Vlastni moduly
16 import fan_pwm as f
17 import sdp6xx as sdp
18
19
20 def read_datetime():
21     '''
22     Vraci cas pristroje.
```

```

23     '''
24     return t.strftime("%y-%m-%d %H:%M:%S")
25
26 def read_pressure(sdp):
27     '''
28     Vraci hodnotu ctení z tlakoveho cidla.
29     '''
30     return sdp.read()
31
32 def new_measurement(pwm_start, pwm_stop, pwm_steps, step_time, name):
33     '''
34     Provadi mereni na zaklade vstupnich parametru.
35     Vraci dataframe obsahujici casove znacky a hodnoty mereni
36     a zaroven jej exportuje do csv souboru.
37     '''
38     # Definovani promennych
39     measurement_name = name
40     dc_list = np.linspace(pwm_start, pwm_stop, pwm_steps)
41     log_df = pd.DataFrame(columns=["datetime", "pressure", "measurement
42                                "])
43
44     measured_data_dict = {}
45     dt = []
46     p = []
47     pin = 12 #Pouzity PWM pin na GPIO sbernici
48     freq = 500 #Frekvence PWM
49
50     # Inicializace senzoru
51     fan = f.Fan(pin = pin, pwm_freq = freq)
52     fan.pwm_start(dc = pwm_start)
53     pres = sdp.SDP6xx()
54     read_pressure(pres)
55
56     # Hlavni cyklus mereni
57     for dc in dc_list:
58         fan.pwm_change(dc)
59         print(f"Duty cycle changed to {dc}")
60
61         t.sleep(step_time)
62
63         dt.append(read_datetime())
64         p.append(read_pressure(pres))
65         print(f"Pressure reading: {p[-1]}")
66         measured_data_dict.update({
67             "datetime":dt,
68             "pressure":p,
69             "measurement":measurement_name
70         })
71         df_temp = pd.DataFrame.from_dict(measured_data_dict)
72
73     #Vytvoreni DataFramu s merenymi daty
74     log_df = log_df.append(df_temp, ignore_index=True)
75     print(log_df, "\n")
76
77     #Ukonceni PWM
78     fan.pwm_stop()

```

```
78
79 #Export CSV s namerenymi daty
80 filename = t.strftime("%y-%m-%d %H_%M_")+measurement_name+"_ "
81 i = 0
82 while os.path.exists(filename+str(i)+".csv"):
83     i += 1
84 filename += str(i) + ".csv"
85 log_df.to_csv(filename)
86
87 return log_df
```

## A.1.2 Skript používaný pro vlastní měření

```
1 """
2 Measurement
3 =====
4
5 Pomoci tohoto skriptu bylo provedeno mereni
6 merneho odporu proti proudeni vzduchu dle CSN EN ISO 9053-1
7 """
8
9 import matplotlib
10 matplotlib.use('TkAgg')
11 import matplotlib.pyplot as plt
12
13 # Vlastni modul
14 import fr_measurement as fr
15
16 # Zmereni tlaku a casu
17 for i in range(1,4):
18     meas = fr.new_measurement(0, 100, 10, 20, f"vzorek_1_{i}")
19
20 # Graficke znazorneni mereni
21 meas.plot(x="datetime", y="pressure", kind="scatter")
22 plt.show()
```

## A.2 Měření doby dozvuku

### A.2.1 Výňatek z modulu `acoustics.room`

```
1  """
2  Room
3  ====
4
5  The room acoustics module contains several functions to calculate
6  the reverberation time in spaces.
7  """
8  import numpy as np
9  import pandas as pd
10
11  from scipy.io import wavfile
12  from scipy import stats
13
14  from acoustics.utils import _is_1d, average
15  from acoustics.signal import bandpass
16  from acoustics.bands import (_check_band_type, octave_low, octave_high,
17                               third_low, third_high)
18
19  SOUNDSPEED = 343.0
20
21  '''
22  .
23  .
24  .
25  vynechany kod
26  .
27  .
28  .
29  '''
30
31
32  def t60_impulse(file_name, bands, rt='t30', method = 'schroeder',
33                  returns = None): # pylint: disable=too-many-locals
34      """
35      Reverberation time from a WAV impulse response.
36
37      :param file_name: name of the WAV file containing the impulse
38      response.
39      :param bands: Octave or third bands as NumPy array.
40      :param rt: Reverberation time estimator. It accepts `t30`, `t20`,
41      `t10` and `edt`.
42      :returns: Reverberation time :math:T_{60}`
43
44      """
45      fs, raw_signal = wavfile.read(file_name)
46      band_type = _check_band_type(bands)
47
48      if band_type == 'octave':
49          low = octave_low(bands[0], bands[-1])
50          high = octave_high(bands[0], bands[-1])
```

```

48 elif band_type == 'third':
49     low = third_low(bands[0], bands[-1])
50     high = third_high(bands[0], bands[-1])
51
52 rt = rt.lower()
53 if rt == 't30':
54     init = -5.0
55     end = -35.0
56     factor = 2.0
57 elif rt == 't20':
58     init = -5.0
59     end = -25.0
60     factor = 3.0
61 elif rt == 't10':
62     init = -5.0
63     end = -15.0
64     factor = 6.0
65 elif rt == 'edt':
66     init = 0.0
67     end = -10.0
68     factor = 6.0
69
70 t60 = np.zeros(bands.size)
71
72 # vytvoreni pandas.DataFrame pro kazdou z velicin vystupu
73 if returns == 'progression' or returns == 'all':
74     df_slope = pd.DataFrame(columns=bands)
75     df_intercept = pd.DataFrame(columns=bands)
76     df_t60 = pd.DataFrame(columns=bands)
77     df_progression = pd.DataFrame(columns=bands)
78     df_abs_signal = pd.DataFrame(columns=bands)
79
80 for band in range(bands.size):
81     # Filtrovani signalu
82     filtered_signal = bandpass(raw_signal, low[band], high[band],
83 fs, order=18)
84     abs_signal = np.abs(filtered_signal) / np.max(np.abs(
85 filtered_signal))
86
87     if method == 'schroeder':
88         # Schroederova integrace
89         sch = np.cumsum(abs_signal[::-1]**2)[::-1]
90         sch_db = 10.0 * np.log10(sch / np.max(sch))
91     elif method == 'average':
92         # Exponencialni prumerovani (dle normy CSN EN ISO 354)
93         ewm = pd.Series(abs_signal)
94         ewm = ewm[::-1].ewm(span=int(fs/1)).mean()[::-1]
95         sch = ewm.to_numpy()
96         sch_db = 20.0 * np.log10(sch / np.max(sch))
97
98     # Linearni regrese
99     # Nalezeni pocatku
100     sch_init = sch_db[np.abs(sch_db - init).argmin()]
101     # Nalezeni konce
102     sch_end = sch_db[np.abs(sch_db - end).argmin()]

```

```

102     # Nalezení samplu začátku a konce
103     init_sample = np.where(sch_db == sch_init)[0][0]
104     end_sample = np.where(sch_db == sch_end)[0][0]
105
106     # Vypocet parametru dozvukove krivky
107     x = np.arange(init_sample, end_sample + 1) / fs
108     y = sch_db[init_sample:end_sample + 1]
109     slope, intercept = stats.linregress(x, y)[0:2]
110
111     # Doba dozvuku (T30, T20, T10 or EDT)
112     db_regress_init = (init - intercept) / slope
113     db_regress_end = (end - intercept) / slope
114     t60[band] = factor * (db_regress_end - db_regress_init)
115
116     if returns == 'progression' or returns == 'all':
117         # Pridani dat do prislusneho DataFramu
118         df_slope[bands[band]] = pd.Series(slope)
119         df_intercept[bands[band]] = pd.Series(intercept)
120         df_t60[bands[band]] = pd.Series(t60[band])
121         df_progression[bands[band]] = pd.Series(y)[::1000]
122         df_abs_signal[bands[band]] = pd.Series(abs_signal[
123             init_sample:end_sample+1])
124
125     if returns == 'progression':
126         return df_progression, df_abs_signal
127     elif returns == 'all':
128         return df_progression, df_slope, df_intercept, df_t60,
129         df_abs_signal
130     else:
131         return t60
132
133     '''
134     .
135     .
136     .
137     .
138     .
139     '''

```

## A.2.2 Skript vlastního měření doby dozvuku

```

1  """
2  T60_measurement
3  =====
4
5  Pomoci tohoto skriptu bylo prováděno měření doby dozvuku
6  dle normy CSN EN ISO 354 metodou prerusovaného sumu.
7  """
8  from IPython.display import clear_output
9  from time import sleep, strftime
10 import matplotlib.pyplot as plt
11 import pandas as pd
12
13 #Vlastní moduly
14 from rt60_measurement.measurement import (
15     multiple_measurement,
16     export_t60_to_csv,
17     export_mean_rt60_progs,
18     calculate_all_rt60,
19     beep
20 )
21 from rt60_measurement.plotter import (
22     config_plot_DP,
23     plot_t60
24 )
25
26 # Definování základních vlastností měření
27 mics = 6
28 sources = 2
29 escape_time = 10
30
31 # Definování okrajových podmínek měření
32 temperature = 20.19
33 RH = 48.9
34 bound_dict = {
35     'temp': [temperature],
36     'RH': [RH]
37 }
38 boundary_df = pd.DataFrame(bound_dict)
39
40 # Definování času a názvu měření
41 time_stamp = strftime("%y-%m-%d_%H-%M")
42 variant = "empty_reference_3_frameless"
43 folder = "data"
44
45 # Vytváření korenového adresáře
46 foldername = (
47     time_stamp + '_' + variant
48 )
49 foldertree = os.path.join(folder, foldername)
50 if not os.path.exists(foldertree):
51     os.makedirs(foldertree)
52
53 # Nastavení vzhledu grafu pro zobrazení dílcích výsledku

```



```

54 config_plot_DP(size='small')
55
56 # Export okrajových podmínek
57 boundary_df.to_csv(os.path.join(foldertree, foldername+"_bound_cond.csv"
58 ))
59
60 # # # Hlavní merici cyklus # # #
61 s = 0
62 while s < sources:
63     m = 0
64     while m < mics:
65         # Dotaz na provedení naměru pro danou konfiguraci
66         inp = input(f"Připraven provést měření pro konfiguraci s{s}m{m}
67         }? [y/N]")
68         if not inp == 'y':
69             if input('Mam ?[y/N]') == 'y':
70                 raise KeyboardInterrupt("You've chosen this way, bro.")
71
72         clear_output(wait=True)
73
74         print(f"Mas {escape_time} sekund na opuštění místnosti.")
75         sleep(escape_time) # Odpočet pro opuštění místnosti
76
77         # Vytvoření cesty pro záznam naměru
78         filename = (
79             time_stamp + '_' +
80             variant + '_' +
81             's' + str(s) +
82             'm' + str(m)
83         )
84         path = os.path.join(foldertree, filename)
85
86         # # Samotné měření
87         multiple_measurement("PinkNoise_50_10000_48k_PCM16_LR.wav",
88         path, 3, 10)
89         beep() # Zvuková signalizace konce měření
90
91         # Export měření do CSV souboru pro účel zobrazení
92         export_t60_to_csv(foldertree, filtered=True, s=s, m=m)
93         # Vytvoření grafu dob dozvuku právě provedených naměrů
94         plt.clf()
95         plot_t60(os.path.join(foldertree, "csv_files/t60"), filtered=
96         True, s=s, m=m)
97         plt.ylim([0,10])
98         clear_output(wait=True)
99         plt.show()
100
101         # Dotaz na provedení měření v další konfiguraci
102         if input('Opakovat měření?[y/N]') == 'y':
103             if input('Opravdu?[y/N]') == 'y':
104                 continue
105
106         m += 1
107     s +=1

```

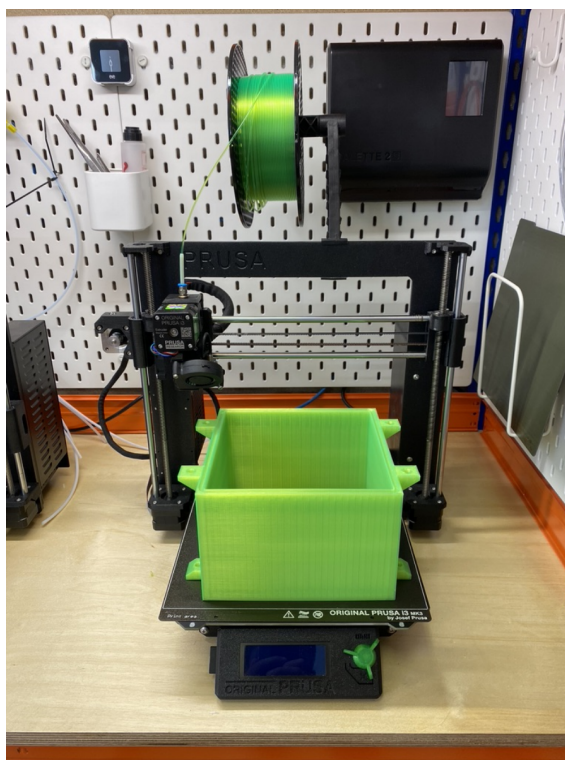


## Příloha B

### Výběr z fotodokumentace

#### B.1 Měření měrného odporu proti proudění vzduchu

##### B.1.1 Měřicí buňka



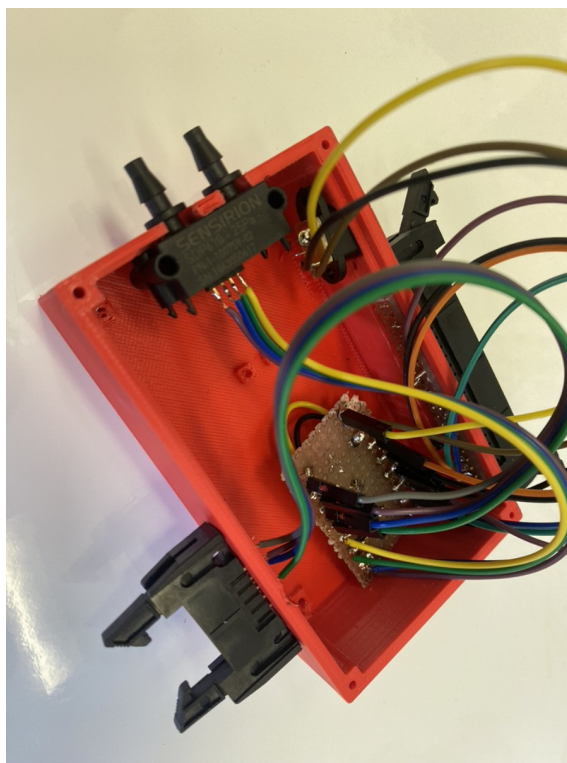
Obr. B.1: Tisk střední části buňky. [autor]



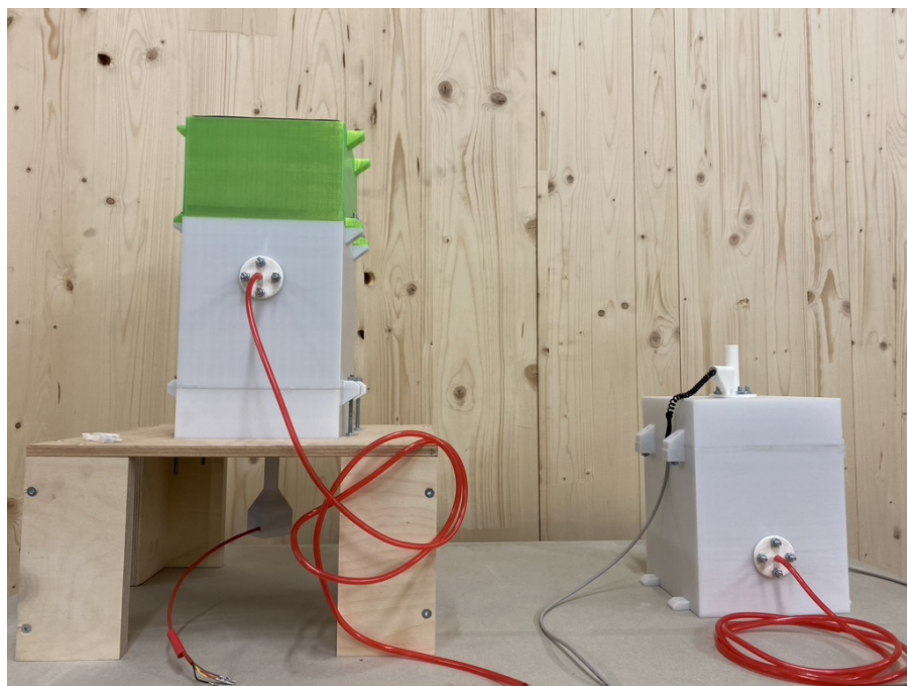
Obr. B.2: Spodní část buňky a její připevnění na stolek. [autor]



Obr. B.3: Pohled do buňky po sešroubování dvou částí. [autor]



Obr. B.4: Pohled do krabičky propojující tlakový senzor, ventilátor a Raspberry Pi.  
[autor]



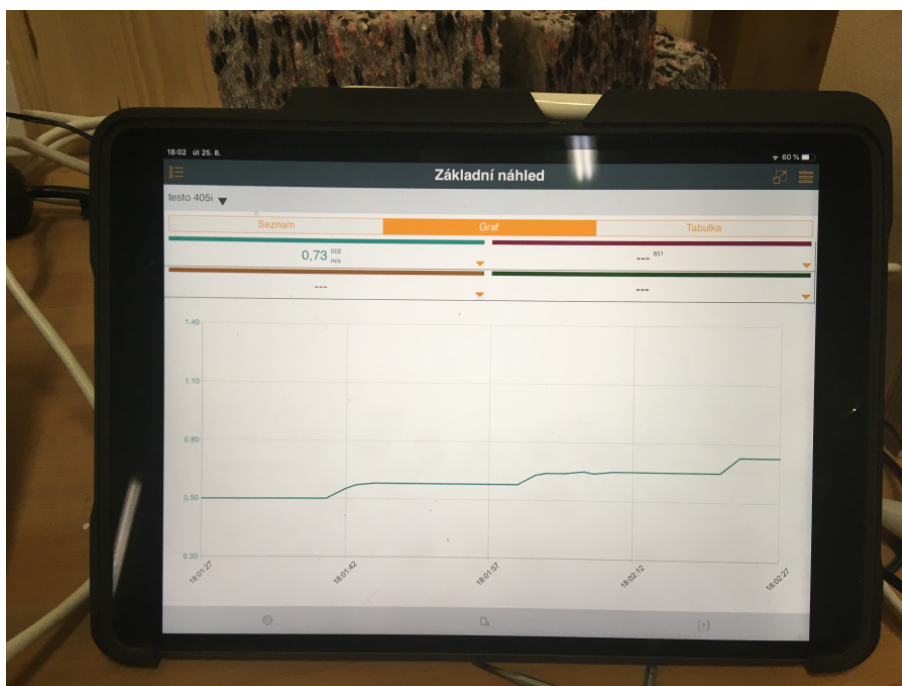
Obr. B.5: Hotová měřící buňka před zapojením elektroniky. [autor]



## **B.1.2 Foto z měření**



Obr. B.6: Pohled do buňky po umístění vzorku. [autor]



Obr. B.7: Uživatelské rozhraní obslužného softwaru k anemometru Testo 405i. [autor]



Obr. B.8: Foto z průběhu měření. [autor]

## B.2 Měření činitele zvukové pohltivosti

### B.2.1 Příprava dozvukové místnosti



Obr. B.9: Úvodní měření dozvuku pomocí balónů. [autor]



Obr. B.10: Úvodní opatření – zakrytí radiátorů. [autor]



Obr. B.11: Výroba rámu pro měření. [autor]





Obr. B.12: Výroba rámu pro měření. [autor]



Obr. B.13: Umístění prvních odrazných desek. [autor]



Obr. B.14: Umístění druhé části odrazných desek. [autor]



Obr. B.15: Měření reference za přítomnosti rámu. [autor]



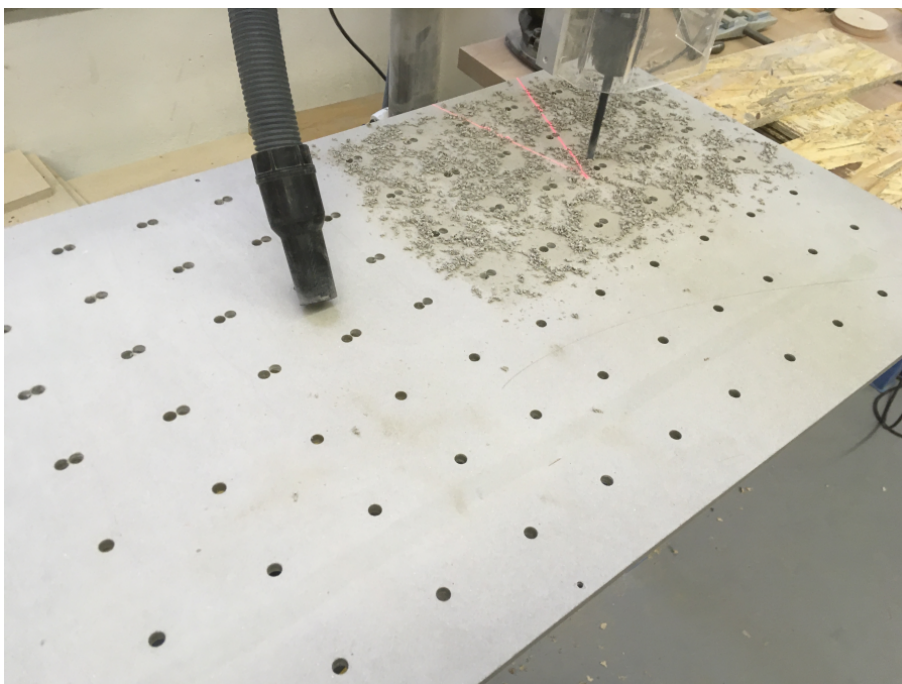


Obr. B.16: Měření reference bez rámu. [autor]

## **B.2.2 Výroba desek**



Obr. B.17: Vrtání děrovaných desek. [autor]

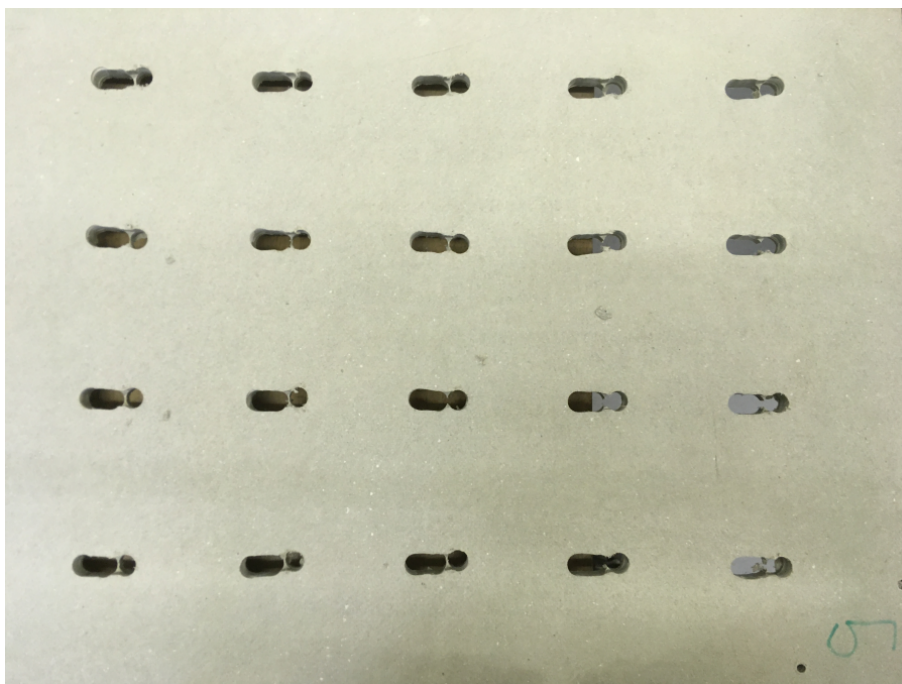


Obr. B.18: Vrtání děrovaných desek, druhá varianta tvaru otvorů. [autor]



Obr. B.19: Děrování desky před profrézováním. [autor]





Obr. B.20: Deska 1\_3 před profrézováním otvorů. [autor]



Obr. B.21: Profrézování souvisejících otvorů. [autor]

### **B.2.3 Z průběhu měření**



Obr. B.22: Desky izolantu připravené pro vložení do rámu. [autor]



Obr. B.23: Izolant umístěný do rámu. [autor]

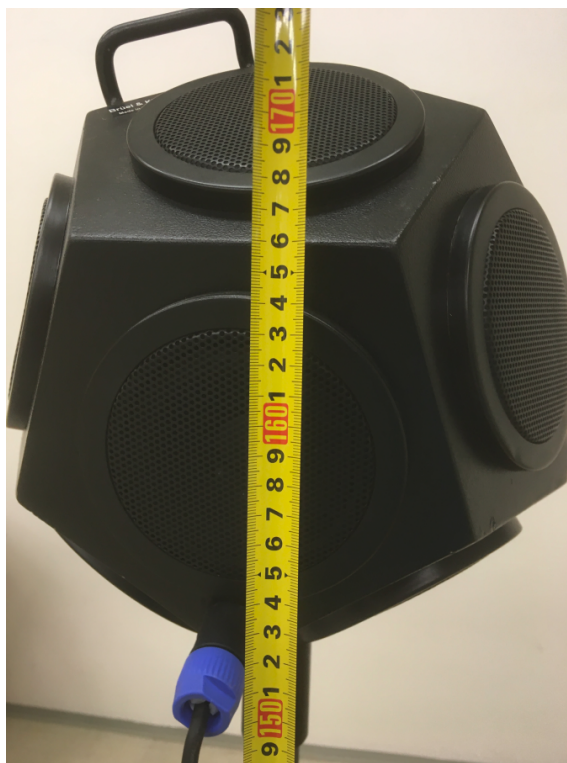


Obr. B.24: Umístění čidla pro měření teploty vzduchu a relativní vlhkosti. [autor]



Obr. B.25: Měřicí sestava - notebook, zvuková karta a výkonový zesilovač. [autor]



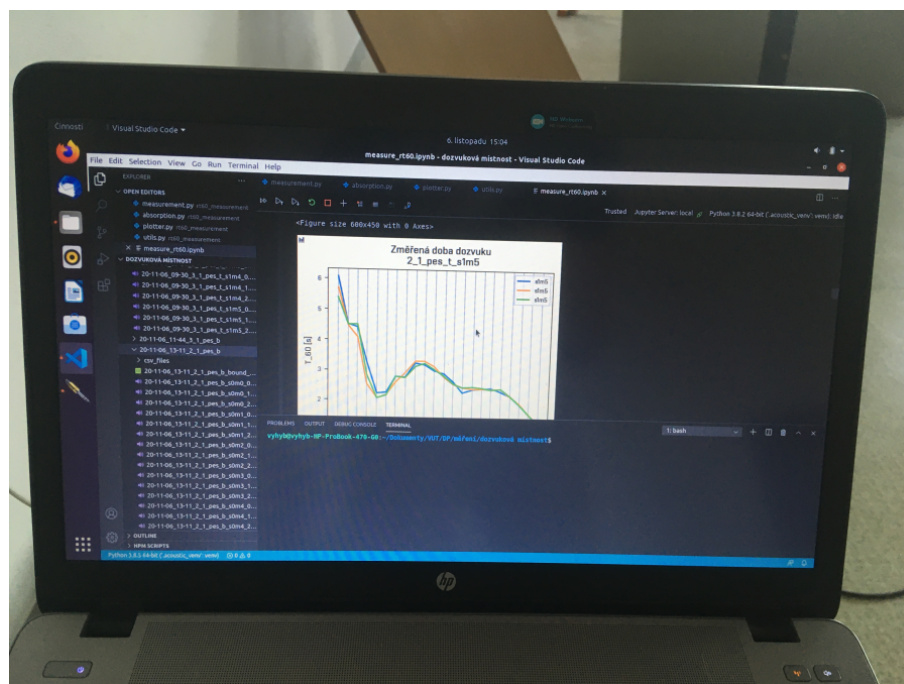


Obr. B.26: Měřící sestava - všesměrový zdroj. [autor]



Obr. B.27: Měřící sestava - všesměrový měřící mikrofon. [autor]





Obr. B.28: Pohled na obrazovku počítače v době měření. [autor]

