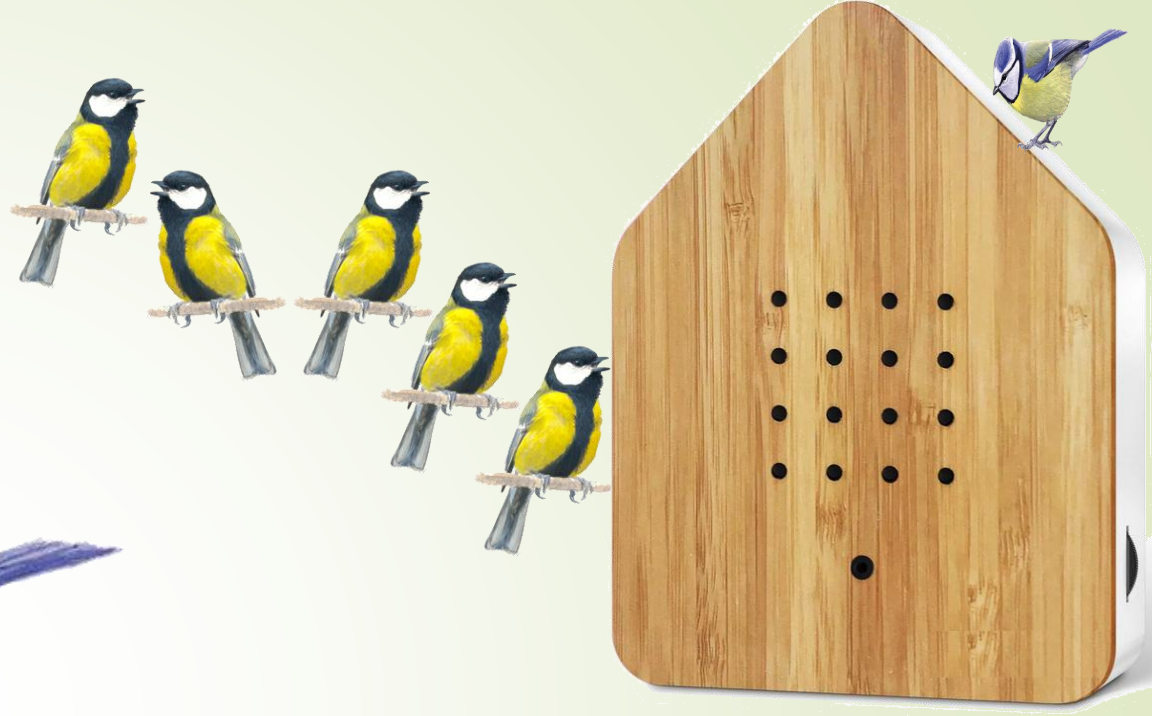






MP3 player

How to create a battery operated, low power, equivalent of a commercially available 'bird-song-relaxing-music-box' to support any MP3 file or file-set.



DEVELOPMENT LAB

Subjects

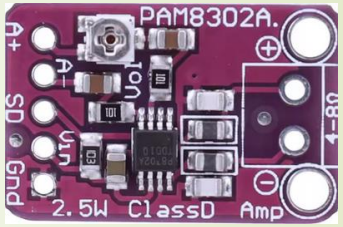
- Hardware design
- Software design
- Case design

DEVELOPMENT LAB

Hardware design: Components / Modules









- Arduino based, low cost modules
- Choice between Arduino Nano or Pololu A-Star 328PB
  - Selected A-Star 328PB (additional UART, Scale factor)
- MP3 module with data storage
  - Module YX5300, which actually contains a YX6300 chip
  - Controllable via UART protocol (RX, TX)
  - Vcc = 3.2V - 5.2V
- Amplifier Class-D MONO
  - Module PAM8302A
  - 2.5W @ 4-8Ω
  - Vcc = 2V - 5.5V
- PIR sensor
  - Module SR602
  - Low standby current (< 20uA)
  - Vcc = 3.3V - 15V
- Power circuit
  - This needs special attention due to the low power requirement

DEVELOPMENT LAB

Hardware design: Power circuit

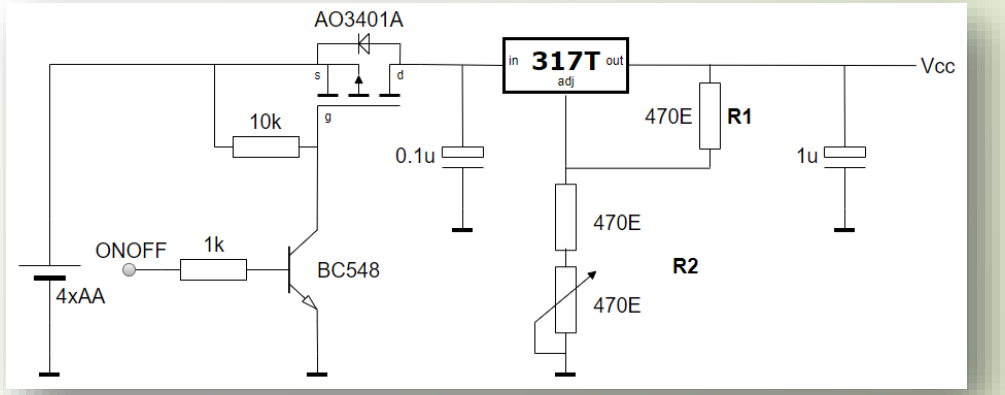
- The CPU and PIR should always be powered
- The CPU will be put in sleep mode, consumes ~200uA
- PIR consumes 20uA, resulting in a total of ~250uA
  Assume 4xAA battery, 2100mAh: 2100mAh/300uA = 7000h = 291 days: Good enough
- Above means we switch the power for the MP3 player and Amplifier off during sleep mode
- We also want to measure the difference between 'power switched off' and 'put MP3 player and amplifier into a sleep or shutdown mode
- Design:
  - The 4xAA, assuming 1.2V/cell = 4.8V, assuming 1.5V/cell = 6V
  - This will be connected to the Vbat input of the A-Star
  - The Vcc output of the A-Star will be connected to the Vcc of the PIR sensor
  - When 1.5V AA cels are used the total voltage level is too high for the MP3 player and amplifier
  - Flexible: add a voltage regulator to be able to adjust Vcc
  - A P-channel MOSFET is put in front of the voltage regulator
  - This MOSFET is controlled via an IO port of the CPU, however this should be save guarded with a transistor to switch the MOSFET gate, this gate rises to Vbat voltage when OFF, this is too high voltage for the IO port

DEVELOPMENT LAB

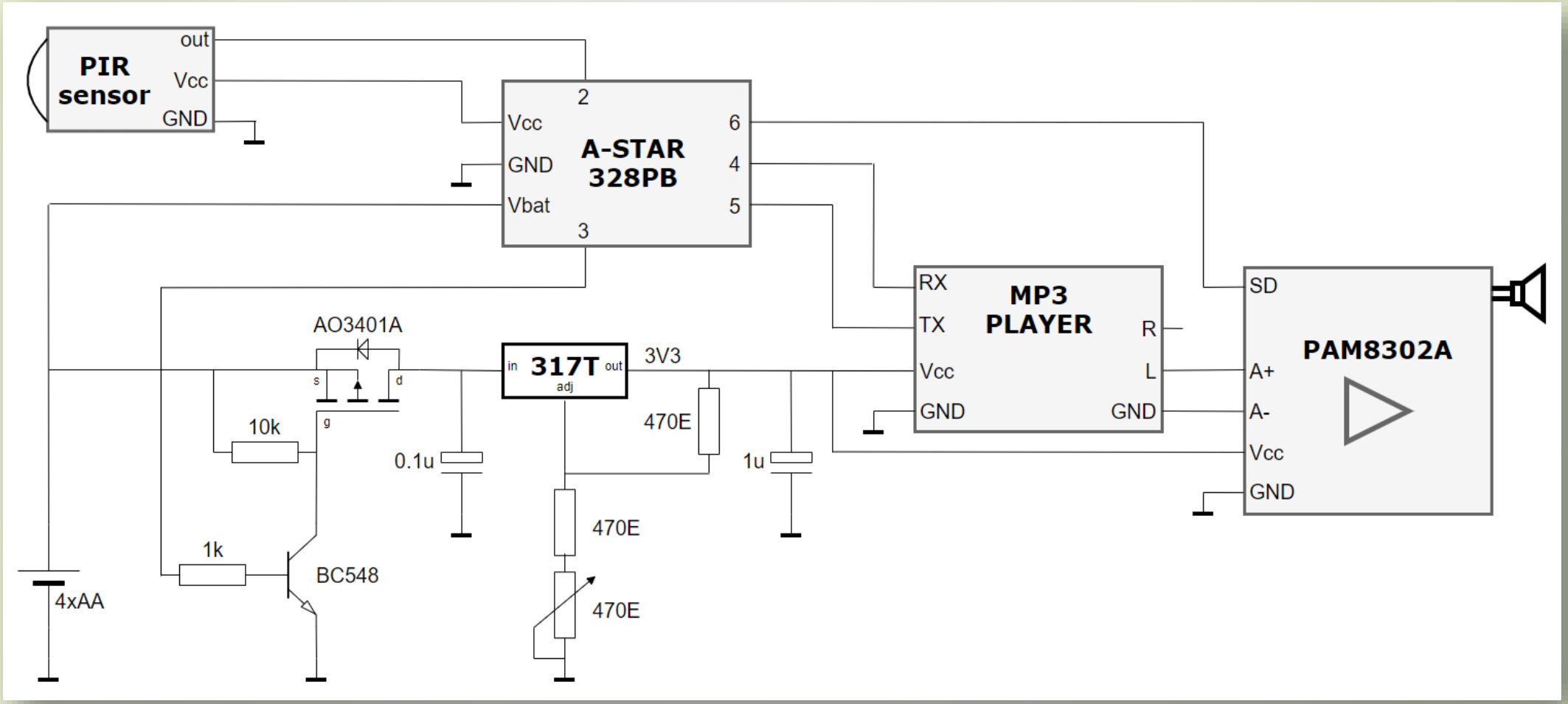
Hardware design: Power circuit

- P Channel MOSFET
  Two types which will work for this application:
  - AO3401A V_{DS}= -30V
  - IRLML6401 V_{DS}= -12V
- Make sure the source is connected to the battery otherwise the diode conducts
- Drain is 0V when the gate is high, equal to Vbat
- Drain is Vbat when the gate is zero:
  This cannot be controlled directly with an IO port of the CPU, since it will put Vbat on the IO port.
  Solution is additional switch by a BC548 NPN transistor or equivalent transistor, this also inverts the logic:
  0 = Power OFF (collector floats)
  1 = Power ON (gate pulled low)

- 317T
  $Vo = 1.25*(1 + R2/R1) + I_{adj}*R2$
  Term "$I_{adj}*R2$" is not significant
  Then: $R2 = R1\ (Vo/1.25 - 1)$
  Choose: $R1 = 470,\ Vo = 3V3 \Rightarrow R2 = 771$
  Assume: $R2min = 470,\ R2max = 940$
  $Vo_min = 2.5V,\ Vo_max = 3.75V$
  Assume: $R2min = 470,\ R2max = 1470$
  $Vo_min = 2.5V,\ Vo_max = 5.16V$

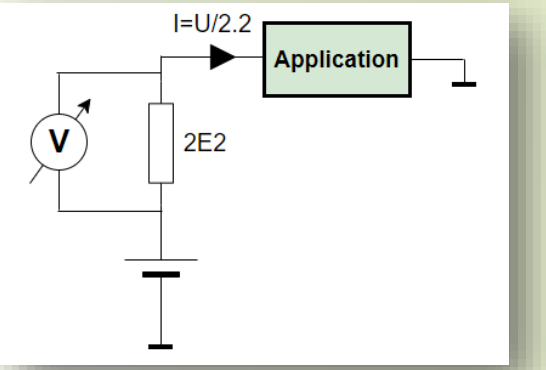


Hardware design: Schematic



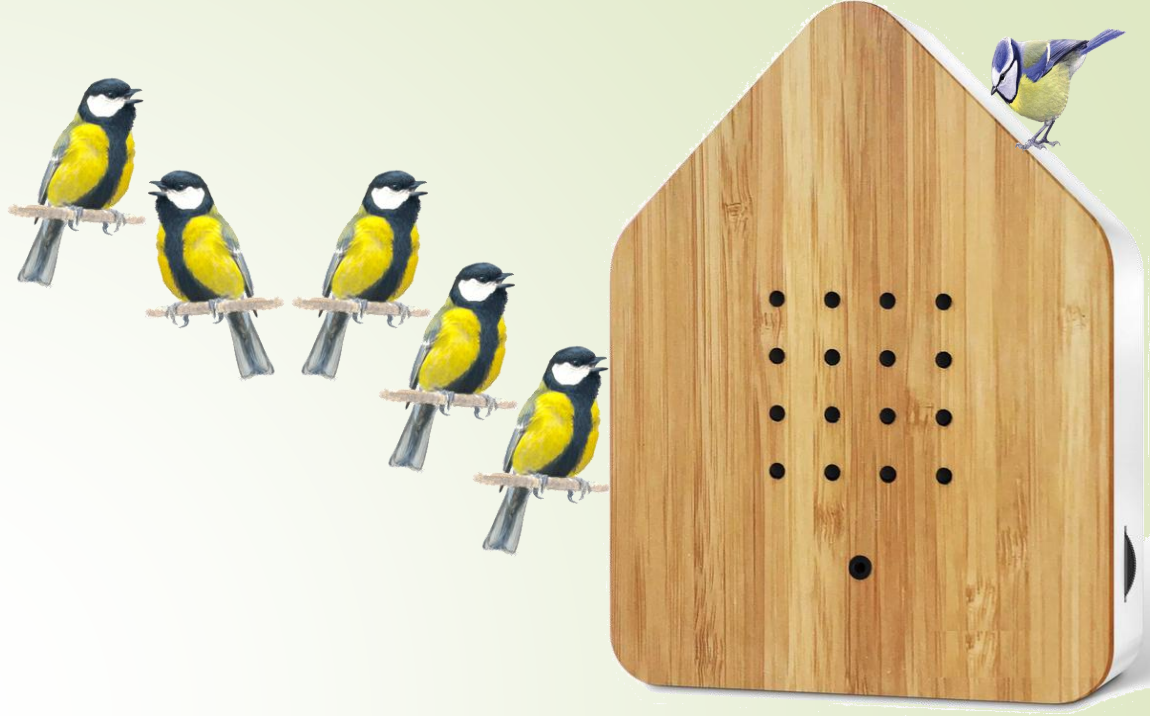
Hardware design: Power measurements

ID	Mode	$V_{R2.2Ohm}$	I_{TOTAL}
1	Power on + Sleep CPU	51.8 mV	23.5 mA
2	Power on + Sleep CPU + Shutdown (MP3/AMP)	44.3 mV	20.1 mA
3	Power on, Playing MP3	60 mV - 90 mV	34 mA
4	Power off + Sleep CPU	0.7 mV	318 µA



- Using the sleep mode and Shutdown mode of the MP3 player and amplifier (ID=2) does not result in low power, consumption is 20.1mA -> 2100mAh / 20.1mA = 104 h = 4.5 days in not active mode and the batteries are empty, thus the switching MOSFET is required
- Switching power off (ID=4), 2100mAh / 318µA = 6600h = 275 d = 9 months sleep
- What would be a realistic usage and how long does the batteries last?
  2100mAh / 34mA = 61 h active usage
  Assume 30mins/day active usage: 61/0.5 = 122 d = 4 months

DEVELOPMENT LAB





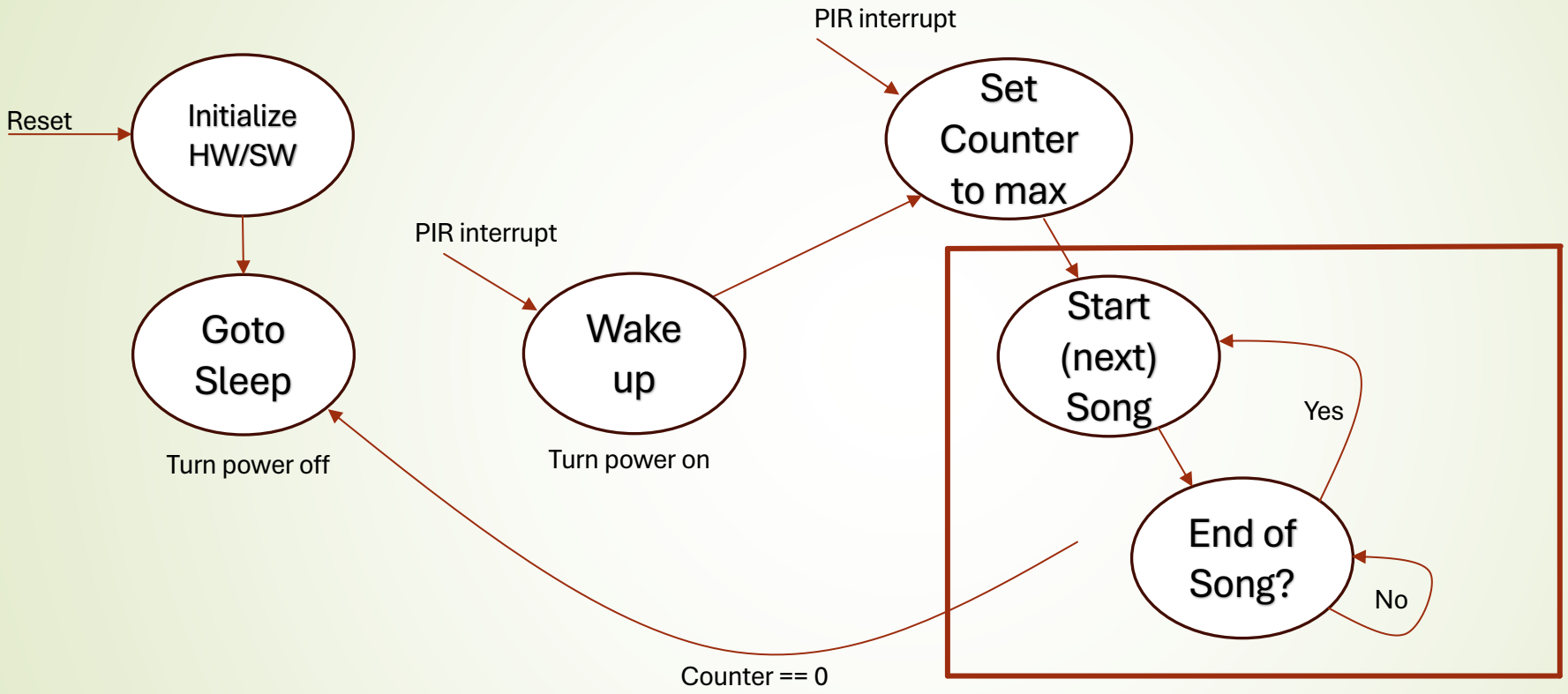
MP3 player, SOFTWARE

- Structure of basic functionality
- Some details



DEVELOPMENT LAB

Software design: Decomposition & Flow



This is the 'active' mode.

Keep in these two states as long as the counter is not terminating.

```
while (counter > 0) {
  // Loop through these two states
  .....
  counter --;
}
```

The PIR interrupt sets the counter to the max value to extend the 'active' mode.

DEVELOPMENT LAB

Software design: Arduino specific

- How does Arduino work related to 'normal' systems
  - Next routine is part of the Arduino core functionality
    File: main.cpp located in the Arduino installation folder

```
#include <Arduino.h>

int main(void) {
    init();
    initVariant();
    setup();

    for (;;) {
        loop();
        if (serialEventRun) serialEventRun();
    }

    return 0;
}
```

There is no magic in software....
Seemingly the Arduino team decided to implement main() themselves.

DEVELOPMENT LAB

Software design: Random playlist

How to create a random playlist

- Assume a list of 10 items, 1 to 10.
  Fetching random numbers and check if they are already used leads to a time consuming solution.
- Ask ChatGPT, Reply: Use the Fischer-Yates Shuffle method

```
int randomList[10];

// Fill the array with numbers from 1 to 10
for (int i=0; i<10; i++) {
     randomList[i] = i+1;
}

// Fisher-Yates Shuffle algorithm
for (int i=9; i>0; i--) {
     // Select a random index between 0 and i
     int j=random(0, i+1);

     // Swap the elements on position i and j
     int tempValue = randomList[i];
     randomList[i] = randomList[j];
     randomList[j] = tempValue;
}
```

DEVELOPMENT LAB

Software design: Interface protocol

Command format (transmit)




```
$S          Start byte, always 0x7e
Version     Version info, always 0xff
Length      Number of bytes of the command without start and end byte
Command     The command code
Feedback    0: no feedback, 1: use feedback (acknowledge)
Data MSB    Data MSB=most significant byte
Data LSB    Data LSB=least significant byte
[csum MSB] Optional 2's complement checksum MSB
[csum LSB] Optional 2's complement checksum LSB
$O          End byte, always 0xef
```

Reply is 10 bytes (receive)

- Same as command format, but with 2 bytes checksum

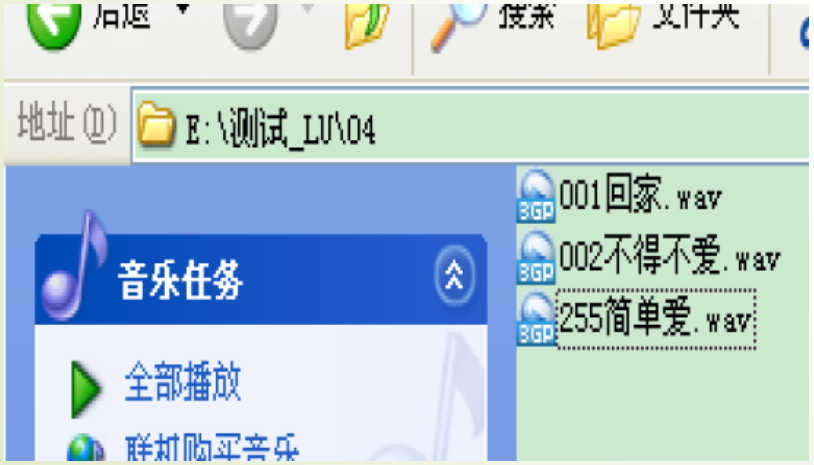
DEVELOPMENT LAB

Software design: Testing, SD content

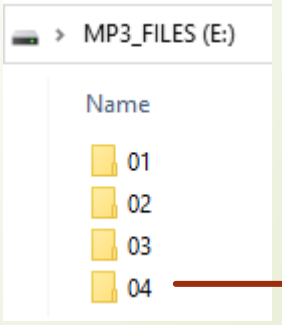
- Using serial interface "print-based testing"
- Using small duration mp3 files
- Content is the spoken number of the file

```
Directory :== <nr><nr>
File      :== <nr><nr><nr><free text>.mp3
```

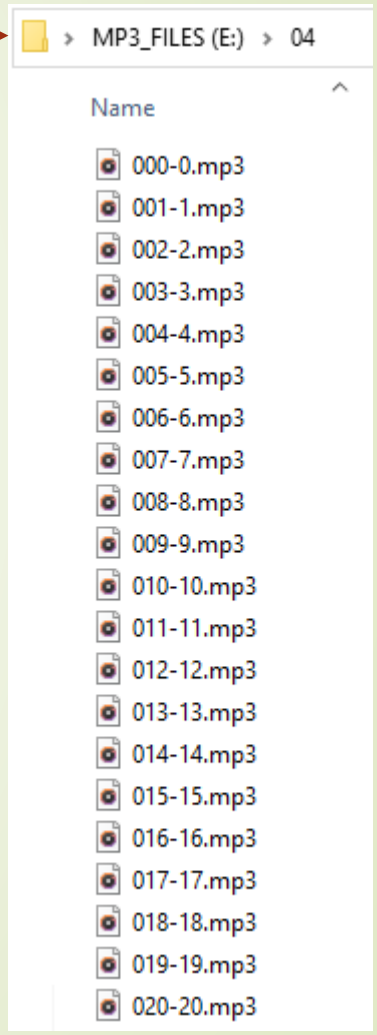
I did not find a real good spec about this format, for instance the filename 01-text.mp3 won't work, make sure to use three numbers



Directories:



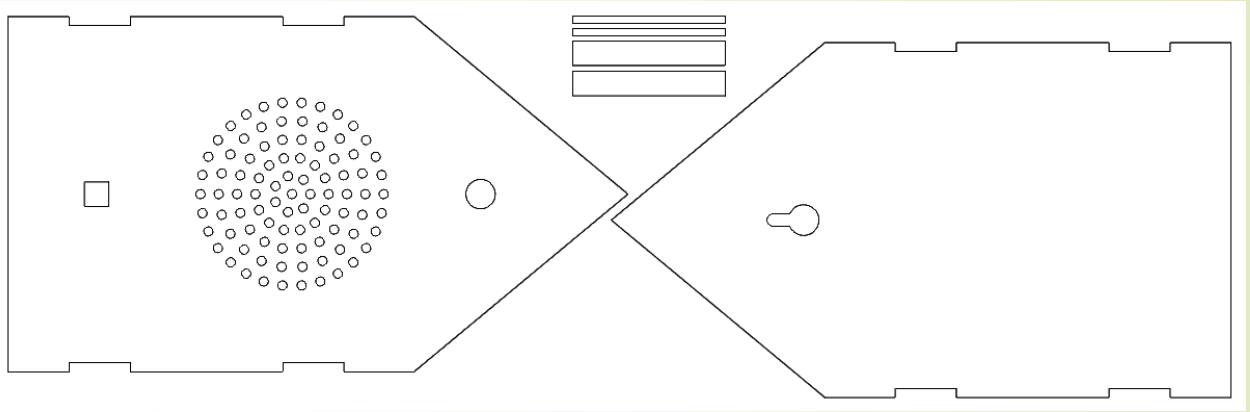
Files:



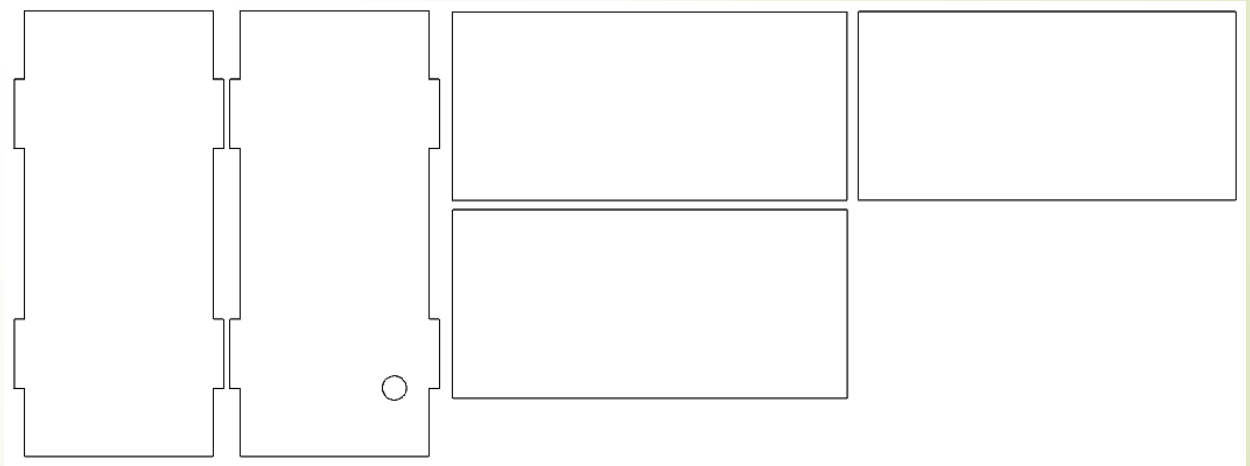
DEVELOPMENT LAB

Case design

- Case is drawn in Fusion
- Parts exported as SVG
- Cut out using xTool S1



- One side is not glued to keep access to the hardware



DEVELOPMENT LAB