Analog Rotary Phones get a Second Life with Raspberry PI

Photo: Hans Gelke

"The horse does not eat cucumber salad"
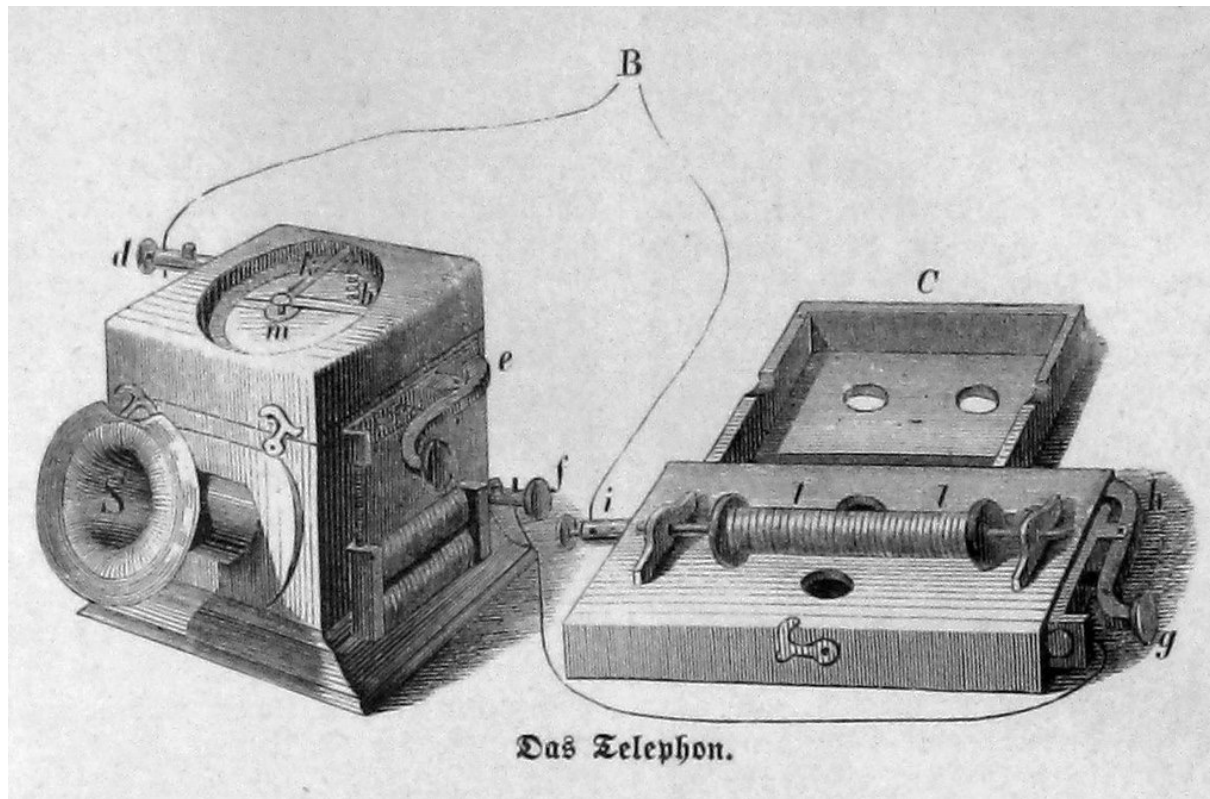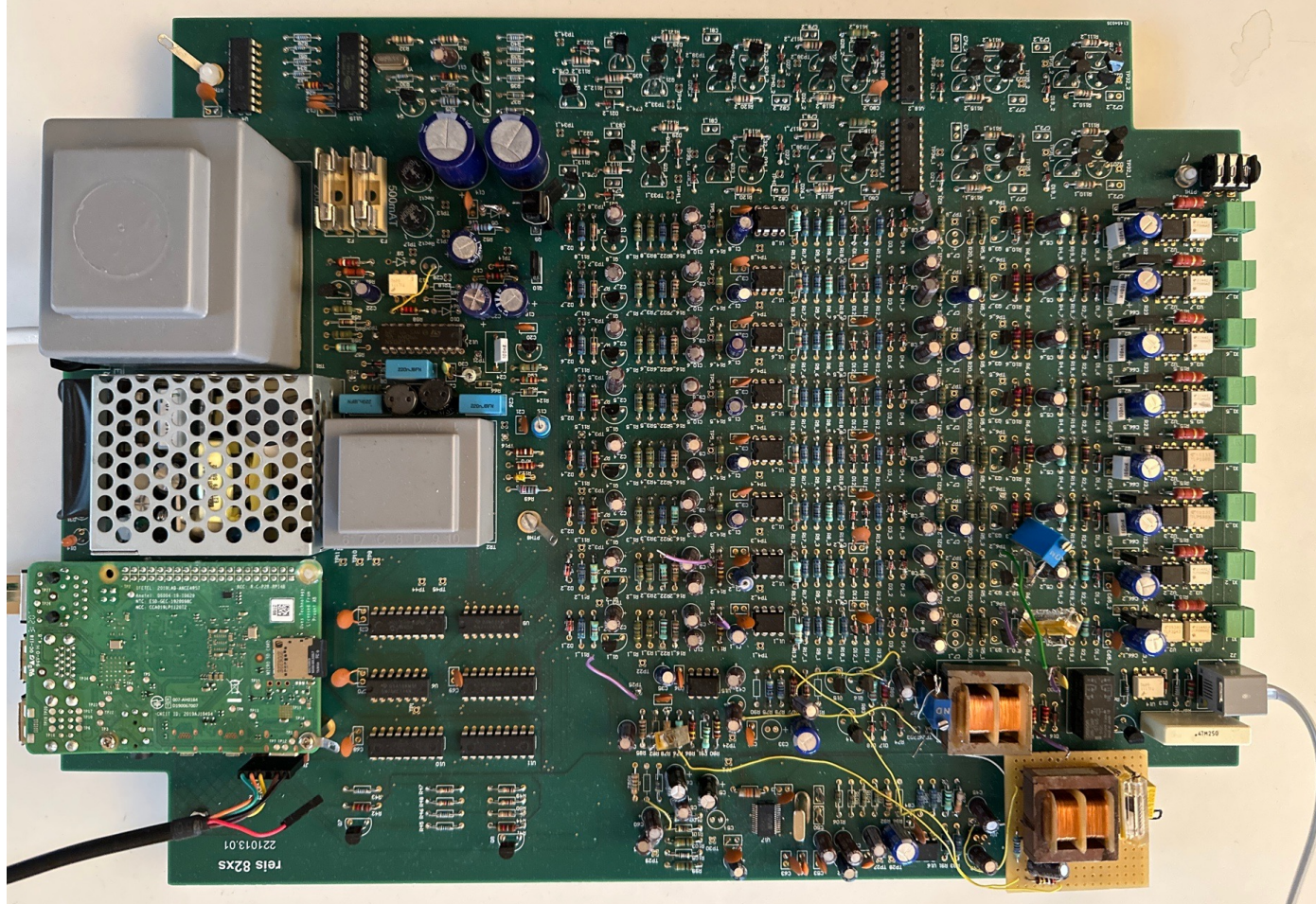

Das Telephon.

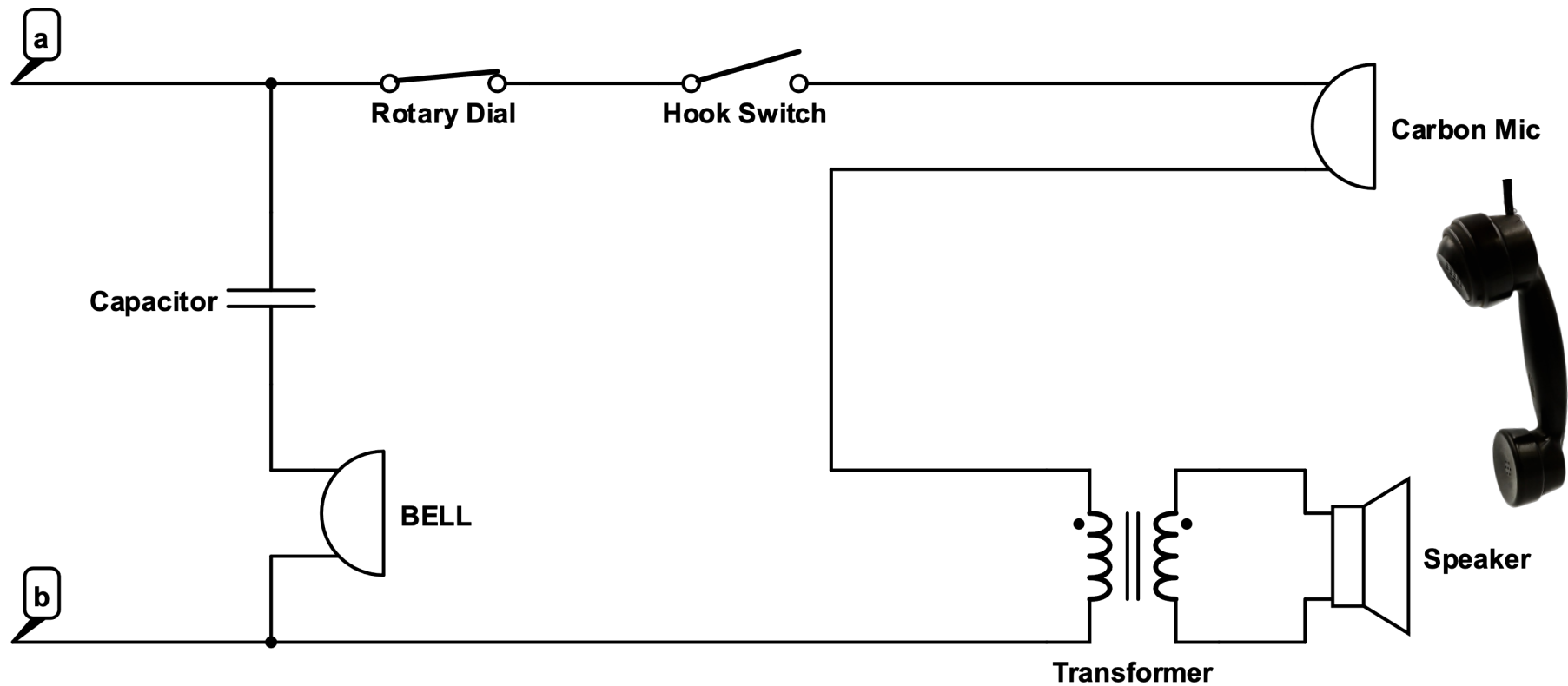Image: Ernst Keil, from the Book : Die Gartenlaube (1863) Page 809

Motivation

Hardware

Functions of Analog Phones

a

b

Rotary Dial

Hook Switch

Carbon Mic

Capacitor

BELL

Speaker

Transformer

Functions of Analog Phones

- Ringing

Voltage at Ringer:
AC 32-100V RMS/25Hz

Functions of Analog Phones

- Idle

a

48V DC

Rotary Dial

Hook Switch

Carbon Mic

Capacitor

BELL

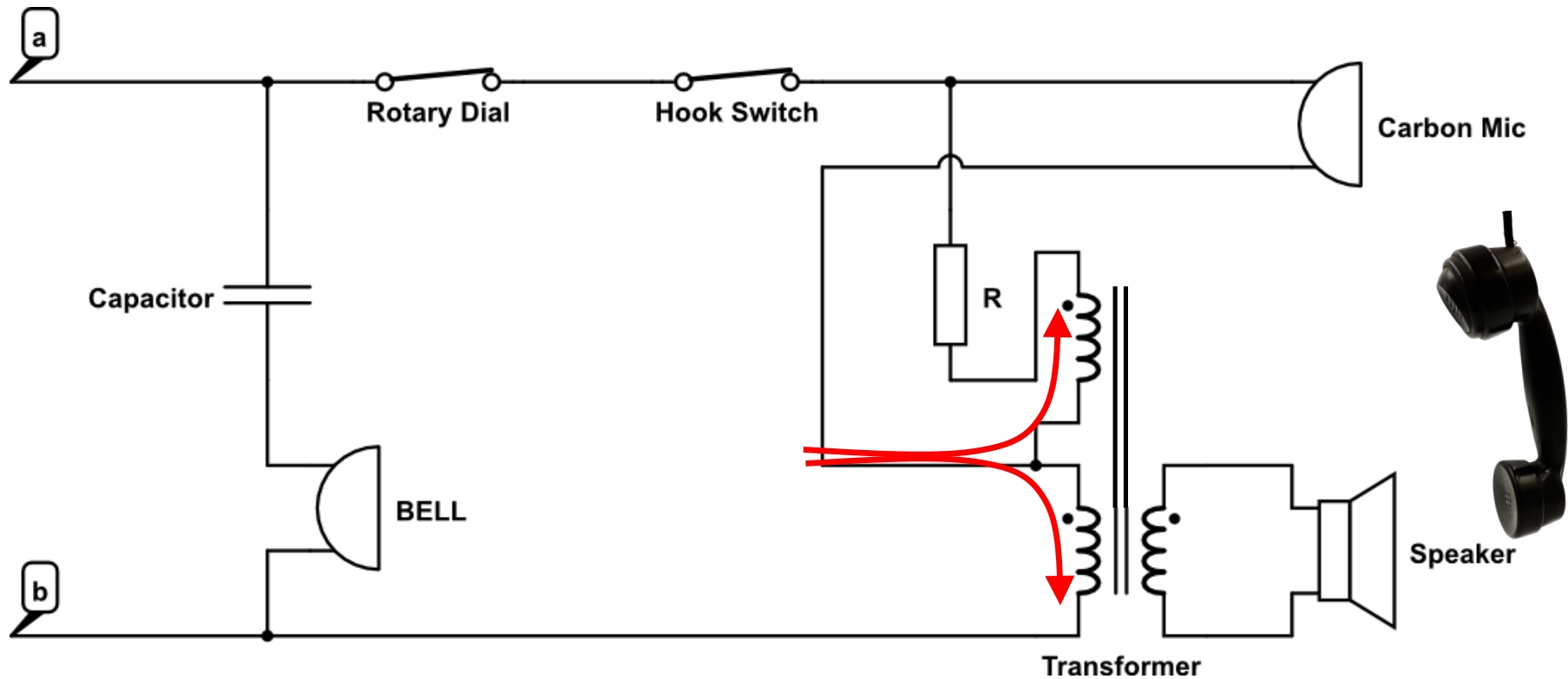**Speaker**

b

Transformer
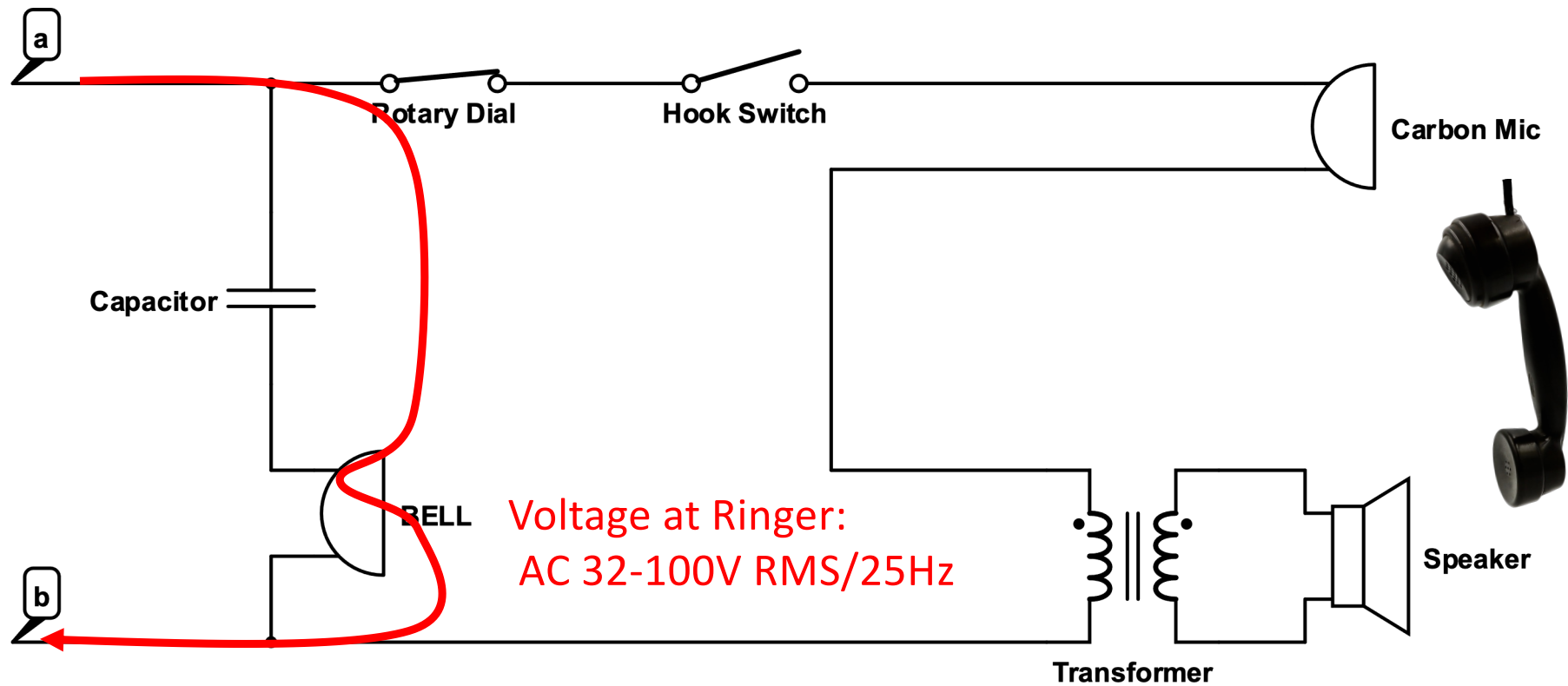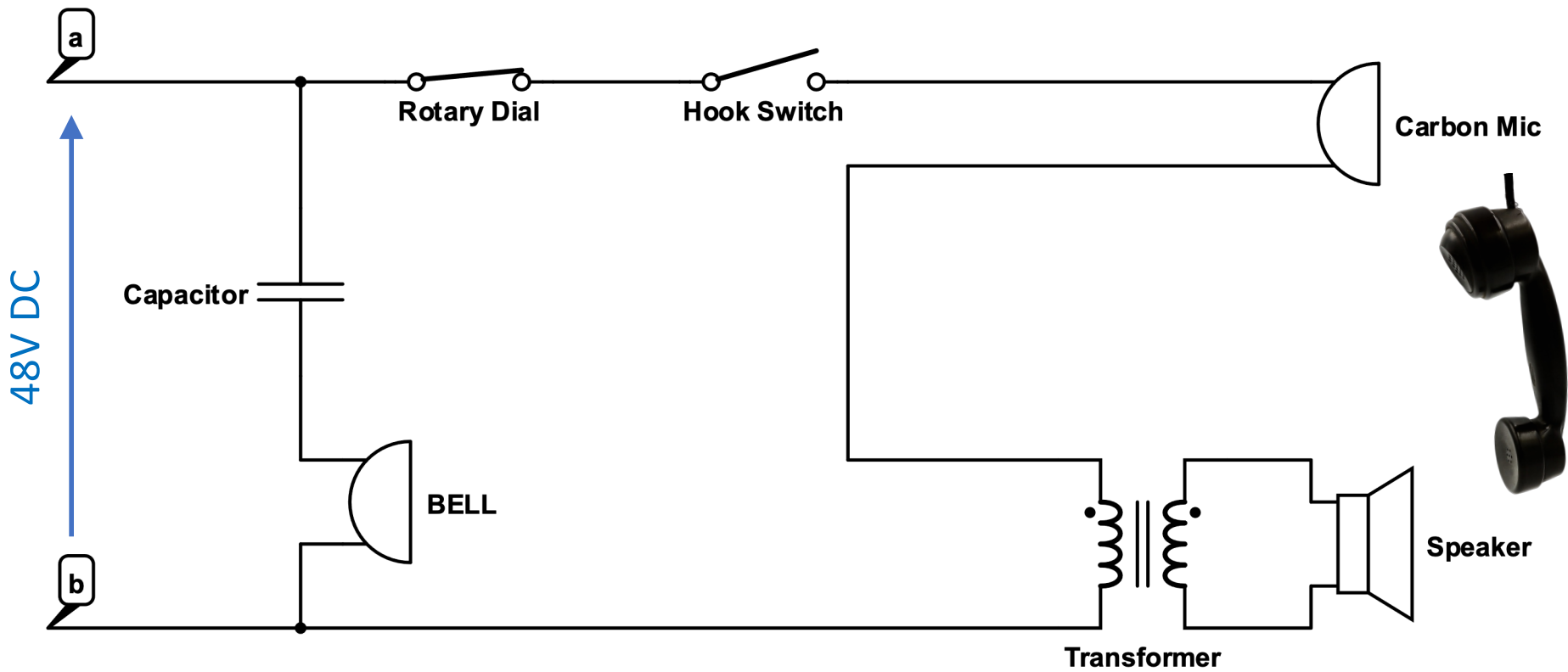
Functions of Analog Phones

- Off Hook
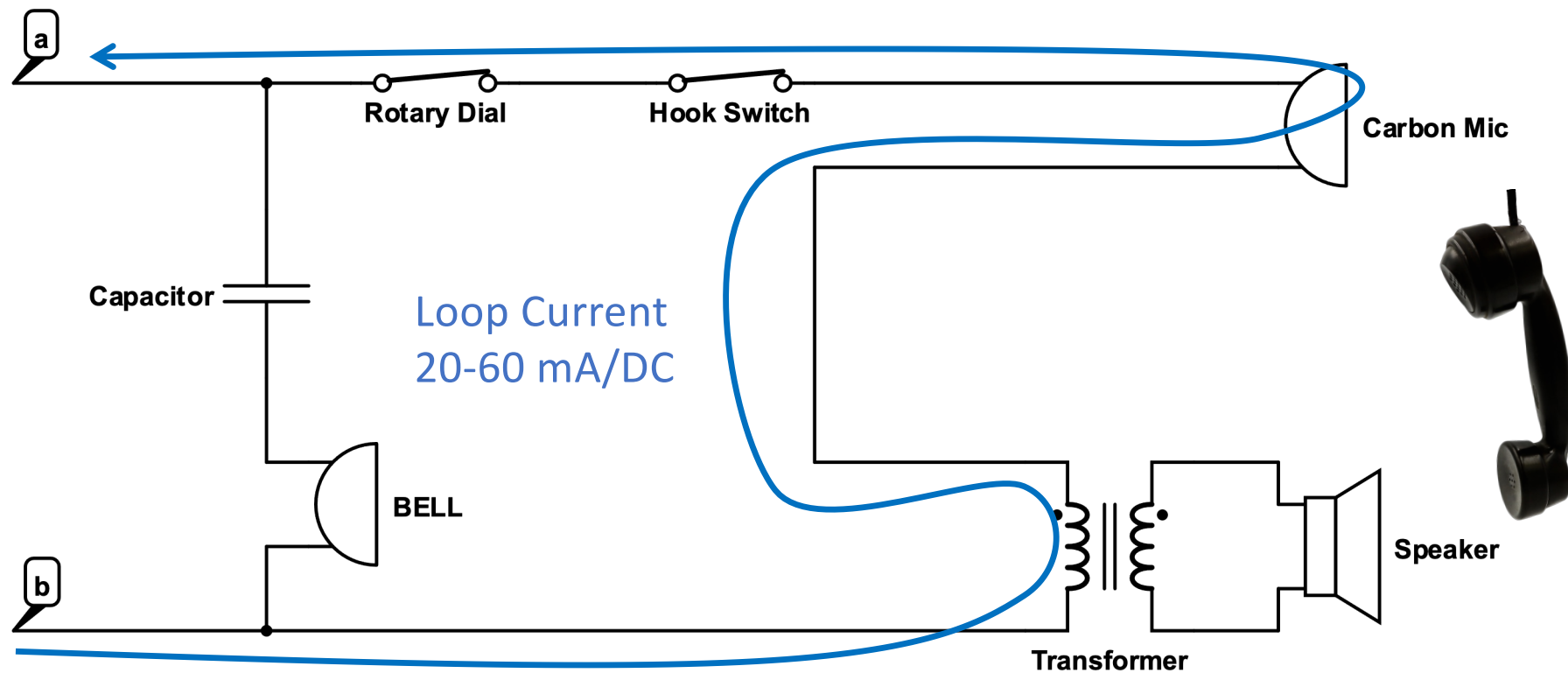
Functions of Analog Phones

- Sidetone Cancelation

Functions of Analog Phones

- Dialing

I_{loop}

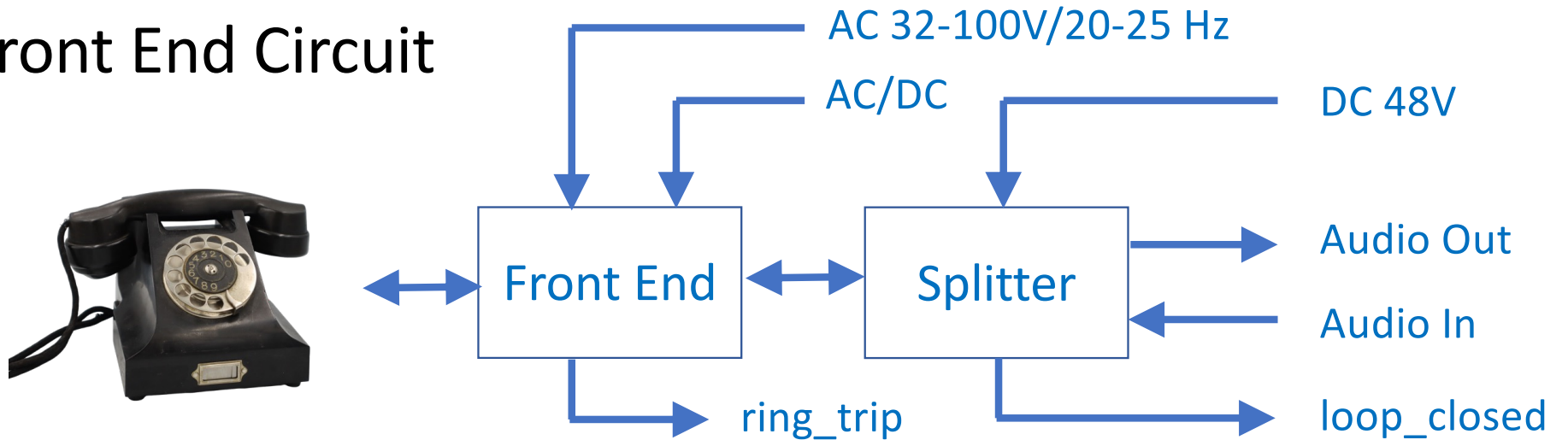100ms

t_{loop}

a

Rotary Dial

Hook Switch

Carbon Mic

Capacitor

Loop Current
20-60 mA/DC

R

BELL

b

Speaker

Transformer
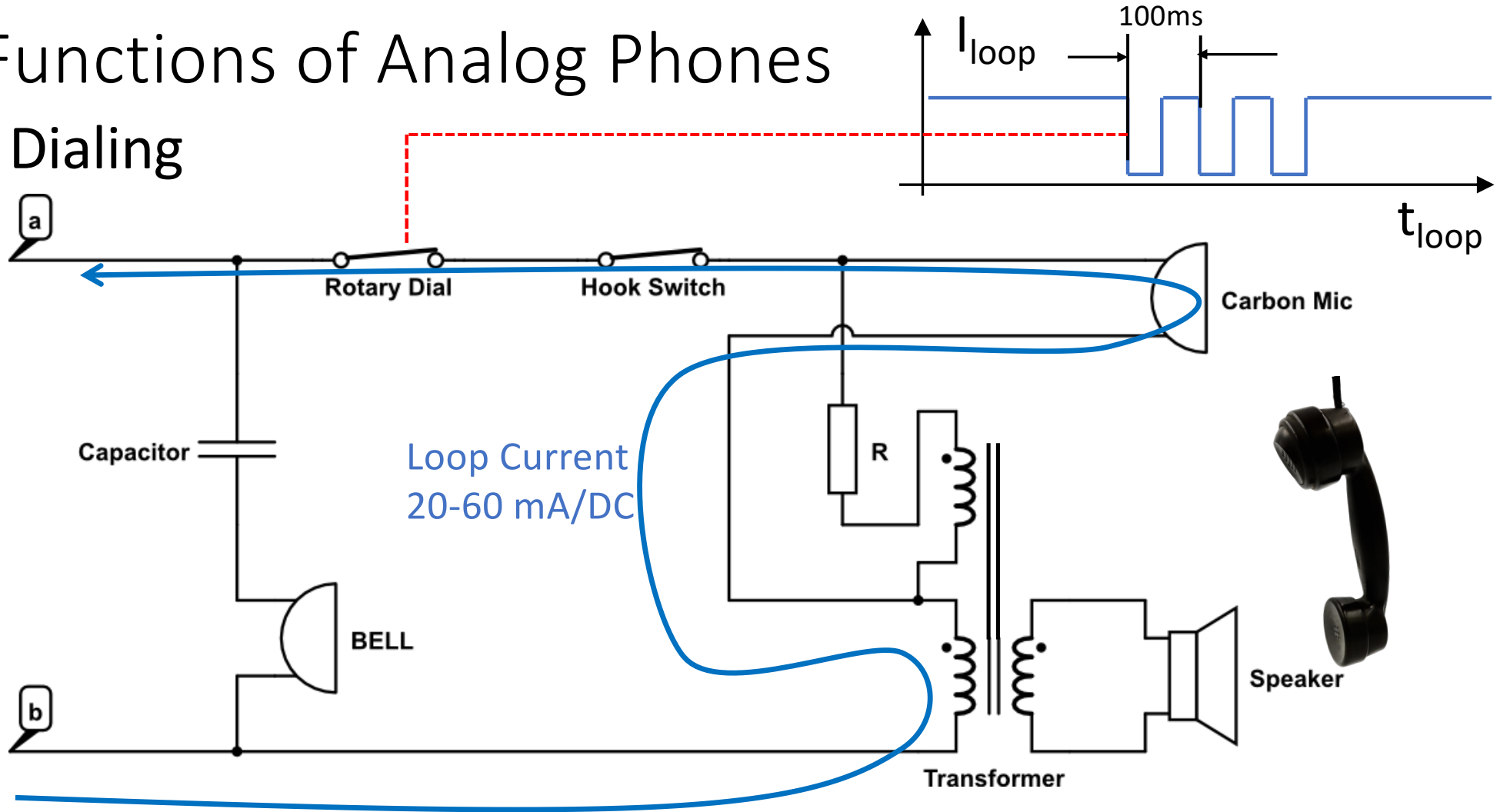
Hardware

• Block Diagram
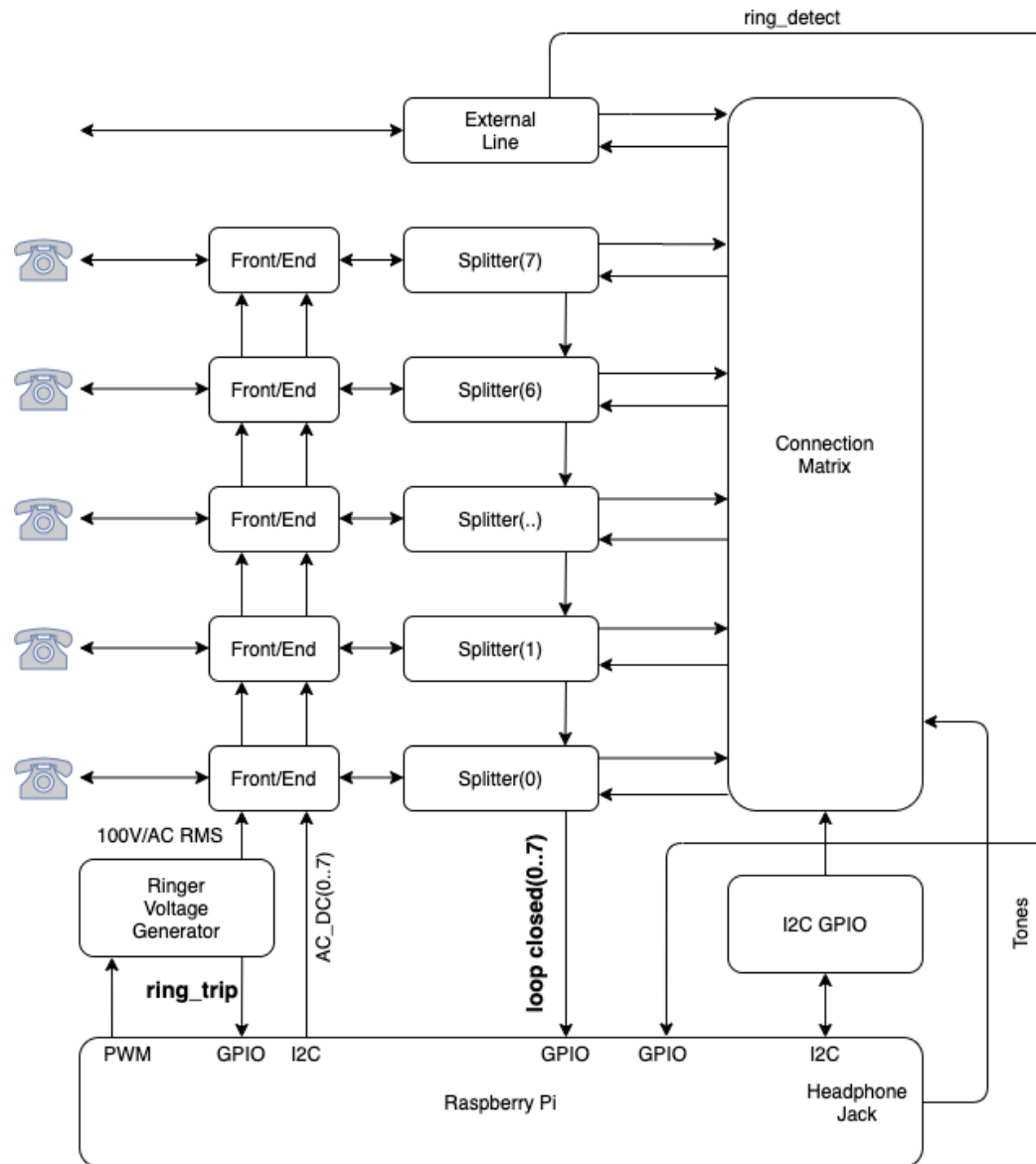
Hardware

- ## Front End Circuit

AC 32-100V/20-25 Hz

AC/DC

DC 48V

Front End

Splitter

Audio Out

Audio In

ring_trip

loop_closed

- Splits full Duplex analog audio into audio_in and audio_out
- Attaches to connection matrix
- Detects when DC-Loop is closed
- Switches to ringing voltage
- Ring Trip

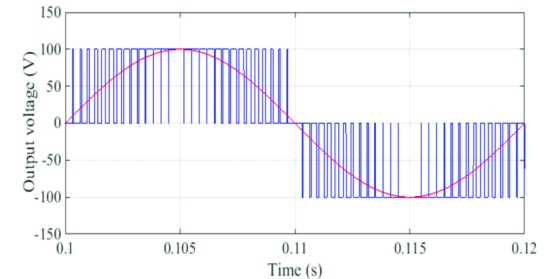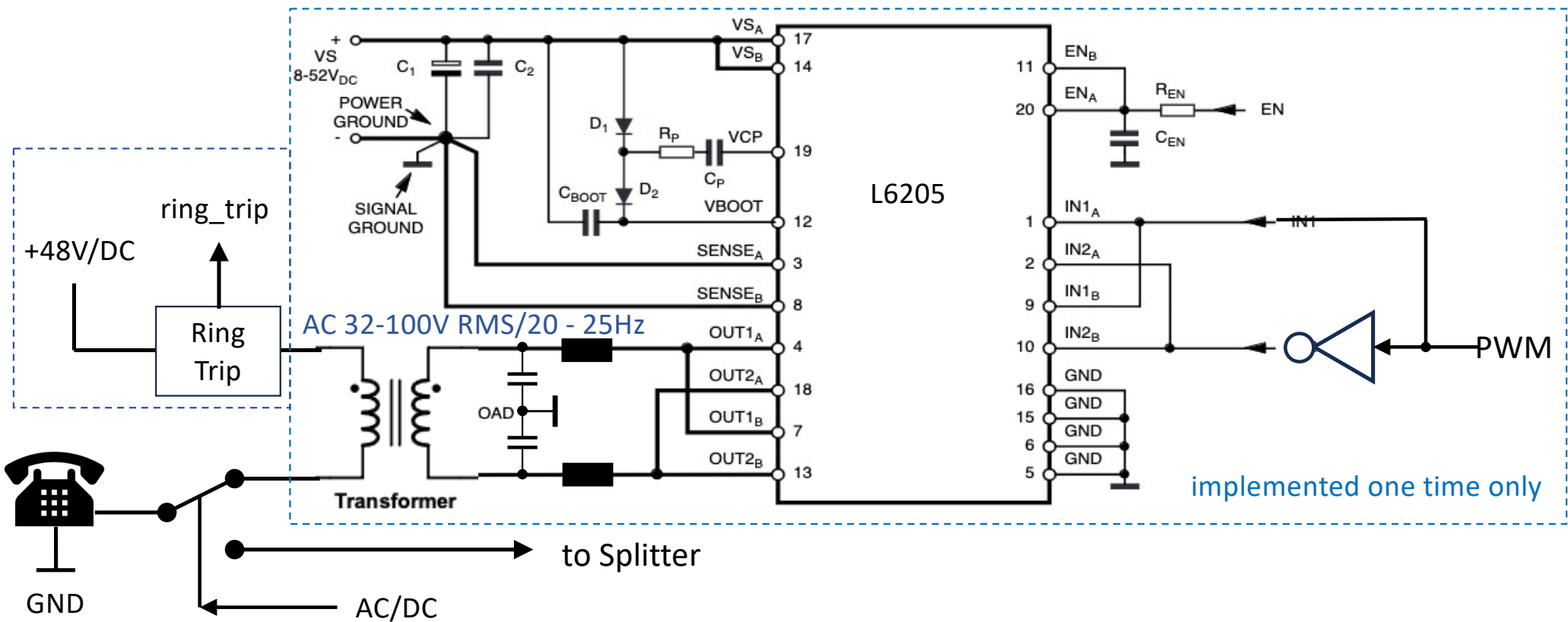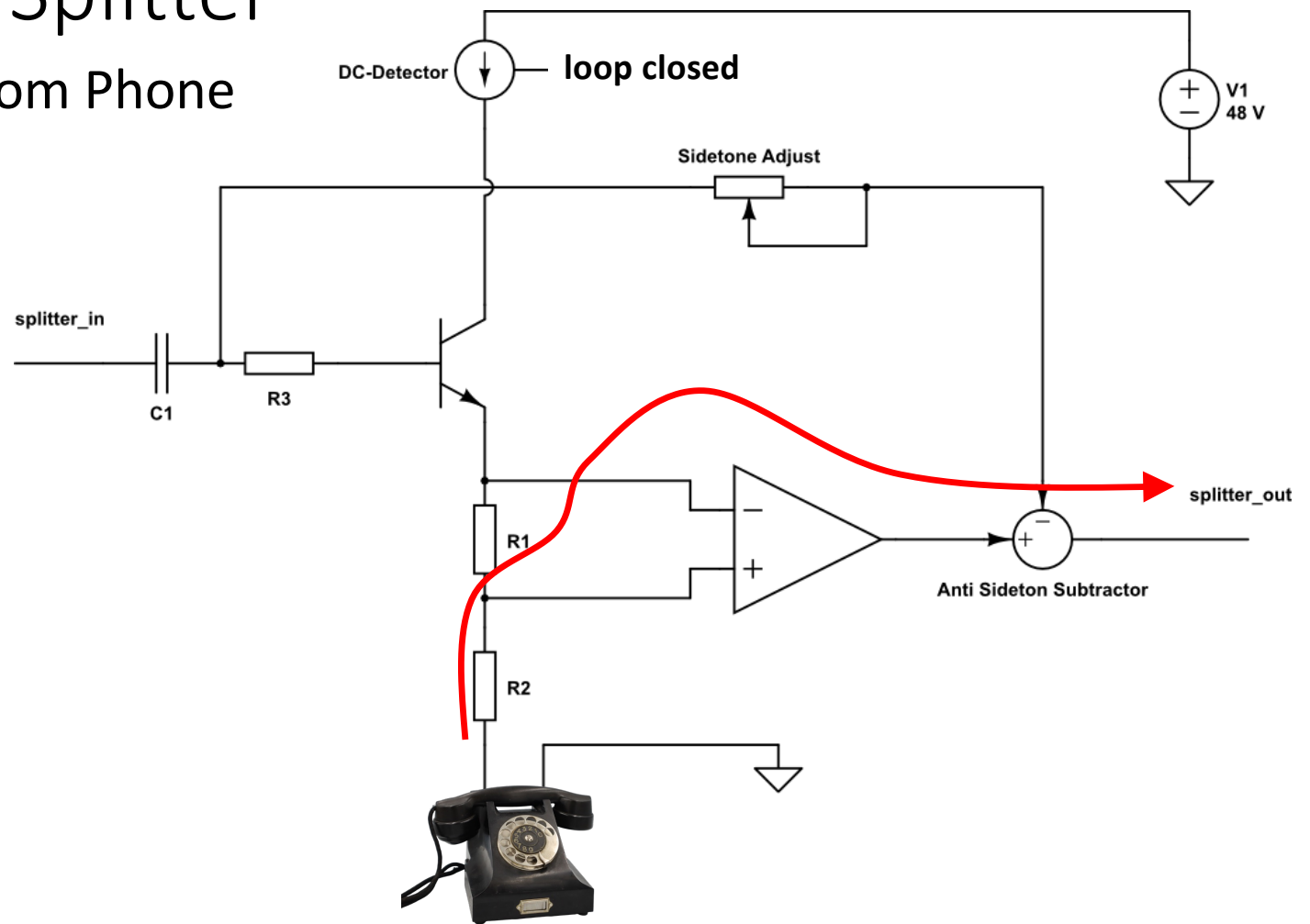# Hardware – Front End

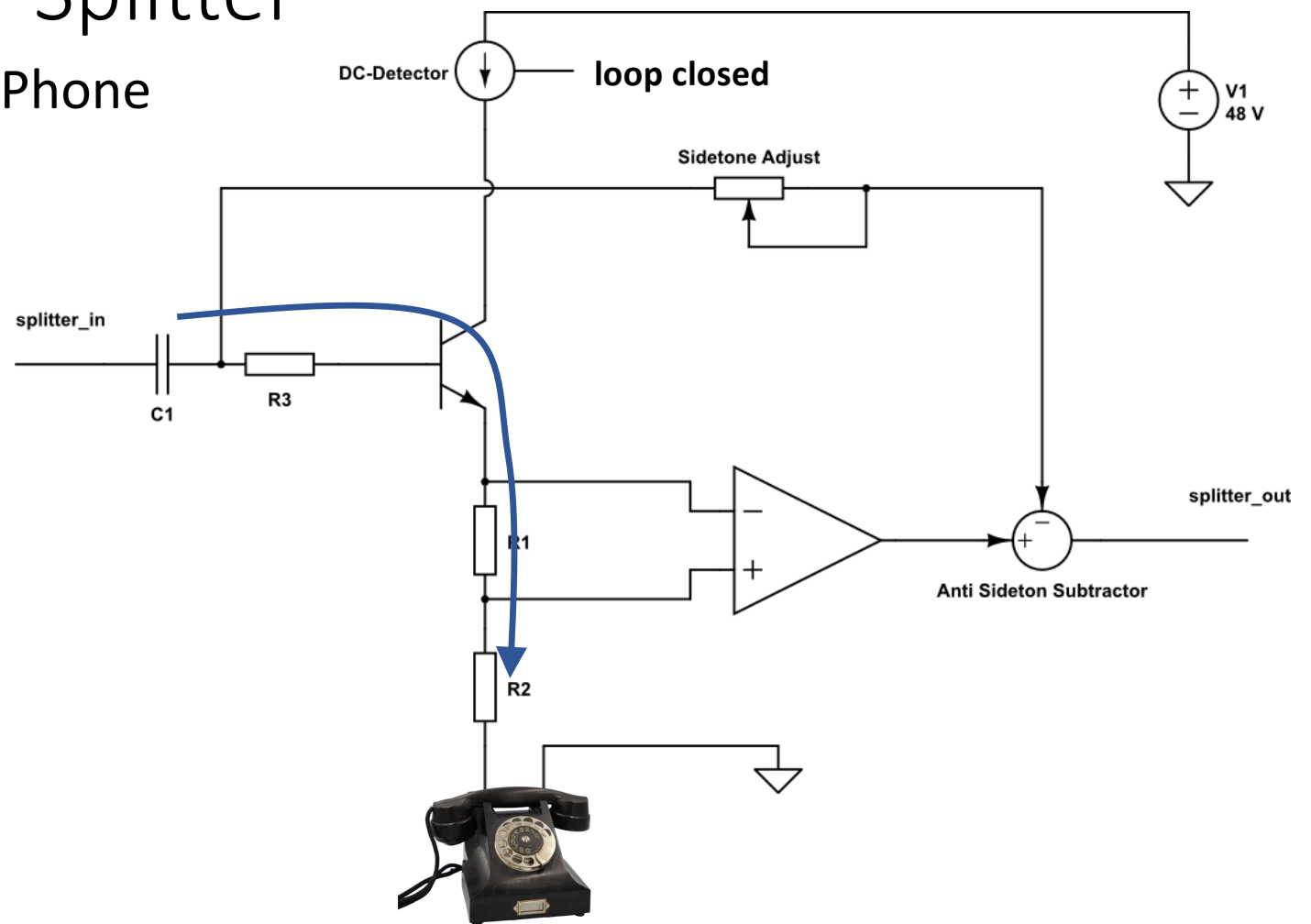- Generating AC Voltage for Bells

Image: Manel Hammami University of Bolognia

AC 32-100V RMS/20 - 25Hz

implemented one time only

Hardware - Splitter

- Audio coming from Phone

Hardware - Splitter

- Audio going to Phone

Hardware - Splitter

- Sidetone Compensation

$V_{splitter\_in}$

splitter_in

C1   R3

DC-Detector   **loop closed**

Sidetone Adjust

V1
48 V

Side Tone
Cancelation

$V_{phone} = V_{splitter\_in} + V_{phone\_in}$

$V_{splitter\_out} = V_{phone} - V_{splitter\_in}$

$V_{splitter\_out} = (V_{splitter\_in} + V_{phone\_in}) - V_{splitter\_in}$

$V_{splitter\_out} = V_{phone\_in}$

R1

R2 V_{phone}

$V_{splitter\_out}$

Anti Sideton Subtractor

V_{phone_in}
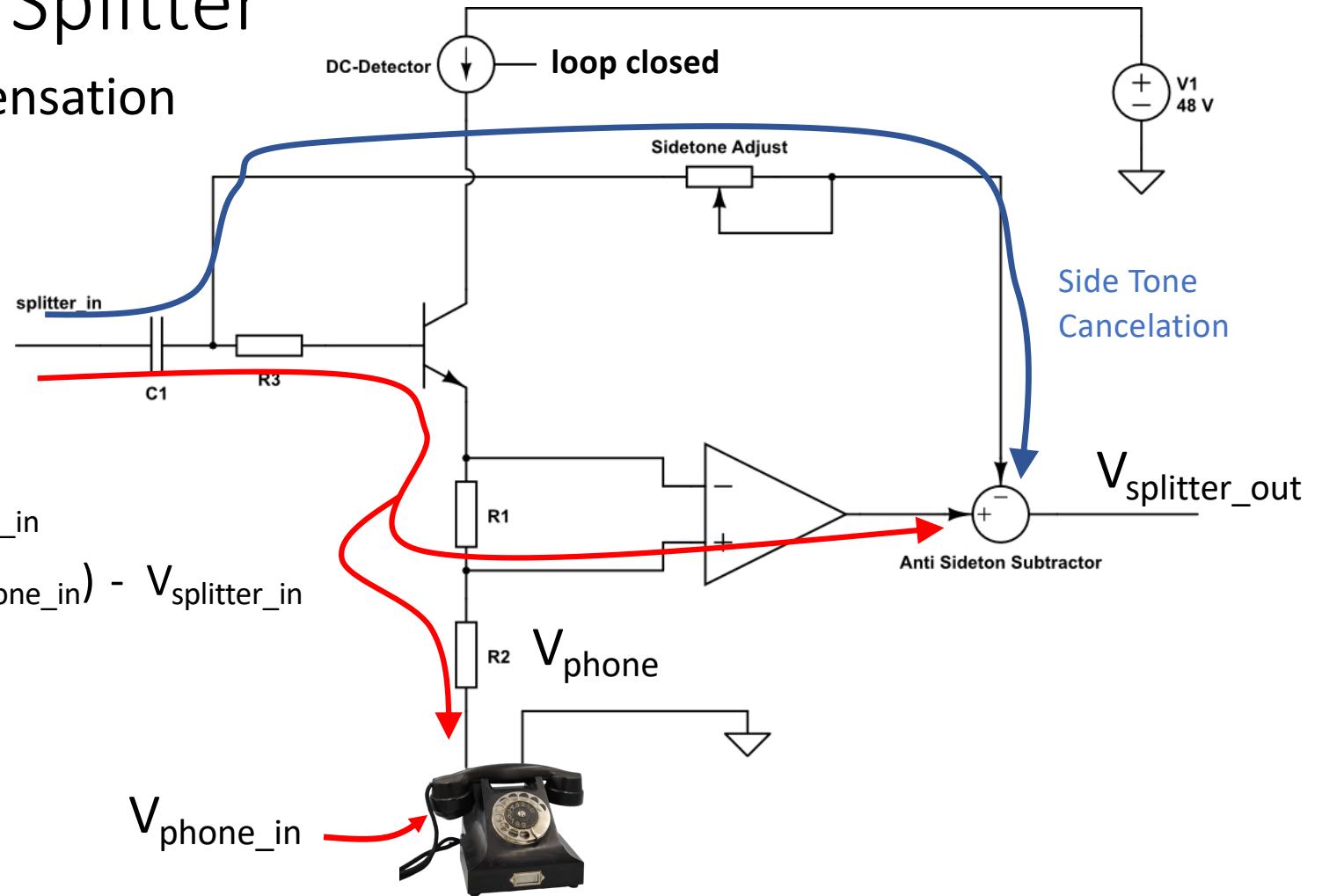
Software

- Overview over realtime tasks

Generates cadences for
ringing, busy etc

generate_signals
sem_wait(&sem_signal)
tonepipeline (GST_STATE_PLAYING)

GST Signals Pipeline

tone_gen
tonepipeline

main_fsm
sem_post(&sem_signal)
ptread_cond_signal(&dial)
pthread_cond_wait(&dialcomple
sem_post(&dtmf)
announcement pipeline (GST_STATE_PLAYING)

rotary
pthread_cond_wait(&dial)
pthread_cond_signal(&dialcomplete)

GST DTMF Pipeline

play_dtmf
sem_wait(&dtmf)

PWM
Frequency

GST Audio Messages Pipeline

play_announcements
announcement pipeline

Task A
Task D
Scheduler →
Task B
Task C
CPU

Round Robbin Scheduling used

Software

- Main FSM

Counting Pulses from Rotary Dialer

Receiver on hook | Receiver off hook | 1. Rotary Pulse | 2. Rotary Pulse | 3. Rotary Pulse | Dial complete | Receiver on hook

FSM states   idle   loop open   loop closed   loop open   loop closed   loop open   loop closed   idle

Dial pulse counter 0   1   2   3
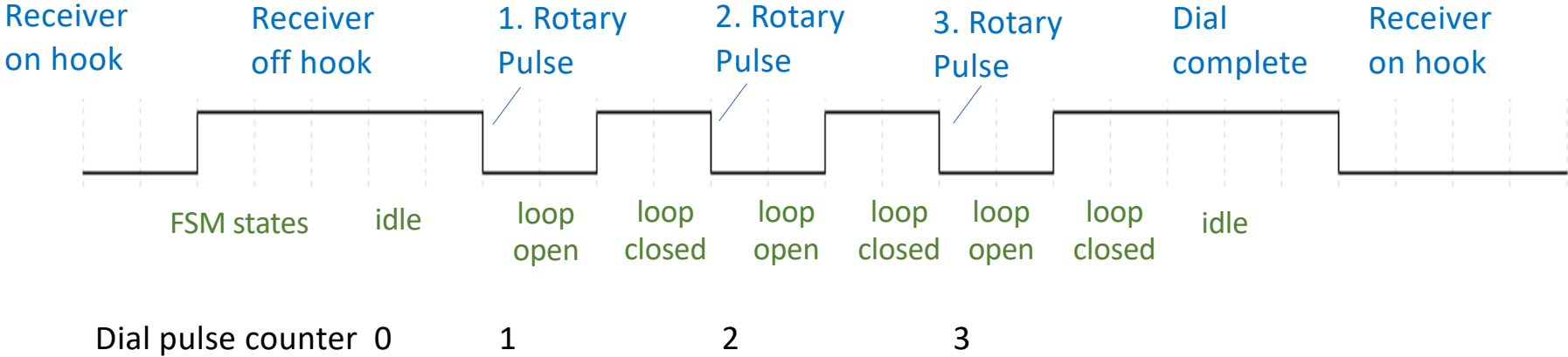
Counting Pulses from Rotary Dialer

- Userspace GPIO exceptions used to count dial pulses (signal **loop_closed**)
- Changes on signal ***loop_closed*** evaluated through pseudo file-system
- ***select()*** allows a program to monitor one or multiple file descriptors
- System call function ***select()*** returns changes in loop_closed files

int select(int *nfds***, fd_set ****readfds***, fd_set ****writefds***, fd_set ****exceptfds***, struct timeval ****utimeout***);**

```
select() > 0 : exception occured
select() == 0  : timeout
select() == -1 : failure
```

This set is watched for "exceptional conditions"

The longest time **select**() may wait before returning, even if nothing interesting happened.

Example:

```
retval = select(my_file +1, NULL, NULL, &fd, &tv);
```
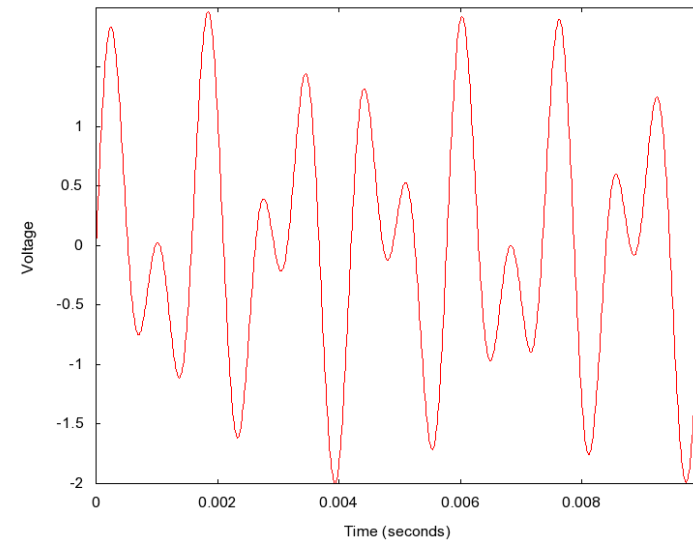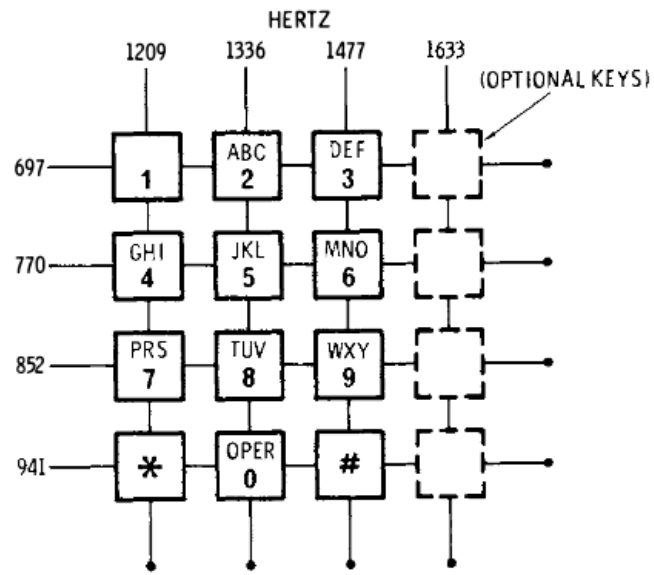
Counting Pulses from Rotary Dialer
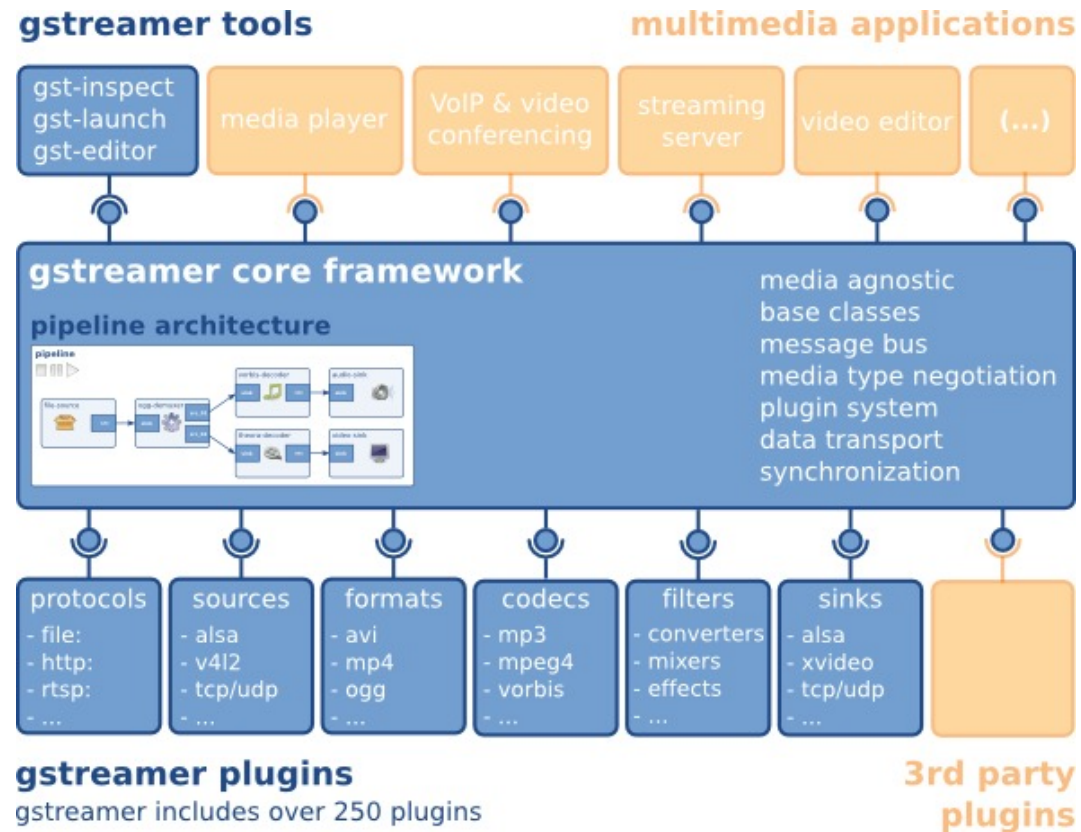
- Rotary Counter - Finite State Machine

Software

• DTMF

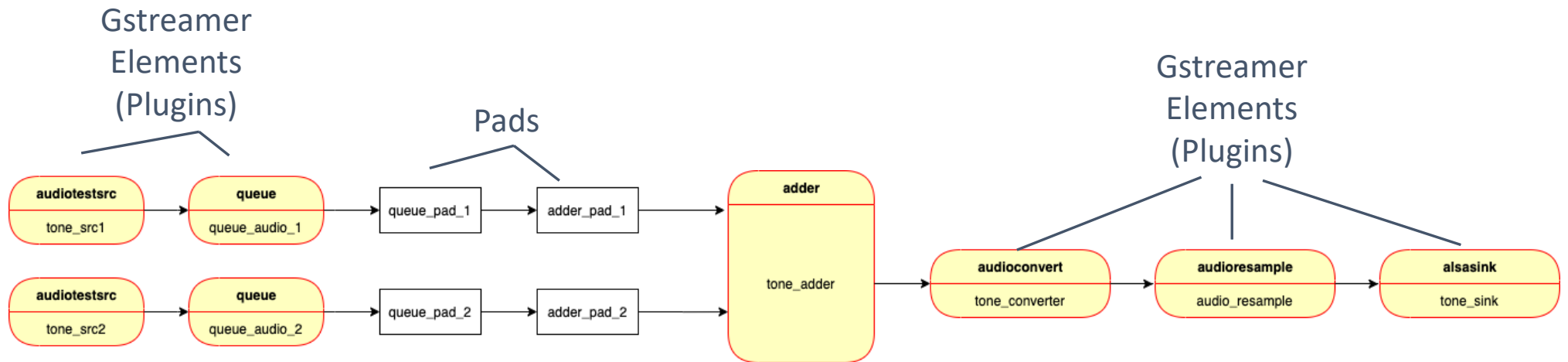Don Lancaster. "TV Typewriter Cookbook". (TV Typewriter). Section "400-Style (Touch-Tone) Modems". p. 177-178.

Software

- GST

Software

- Generating DTMF Pulses and Tones

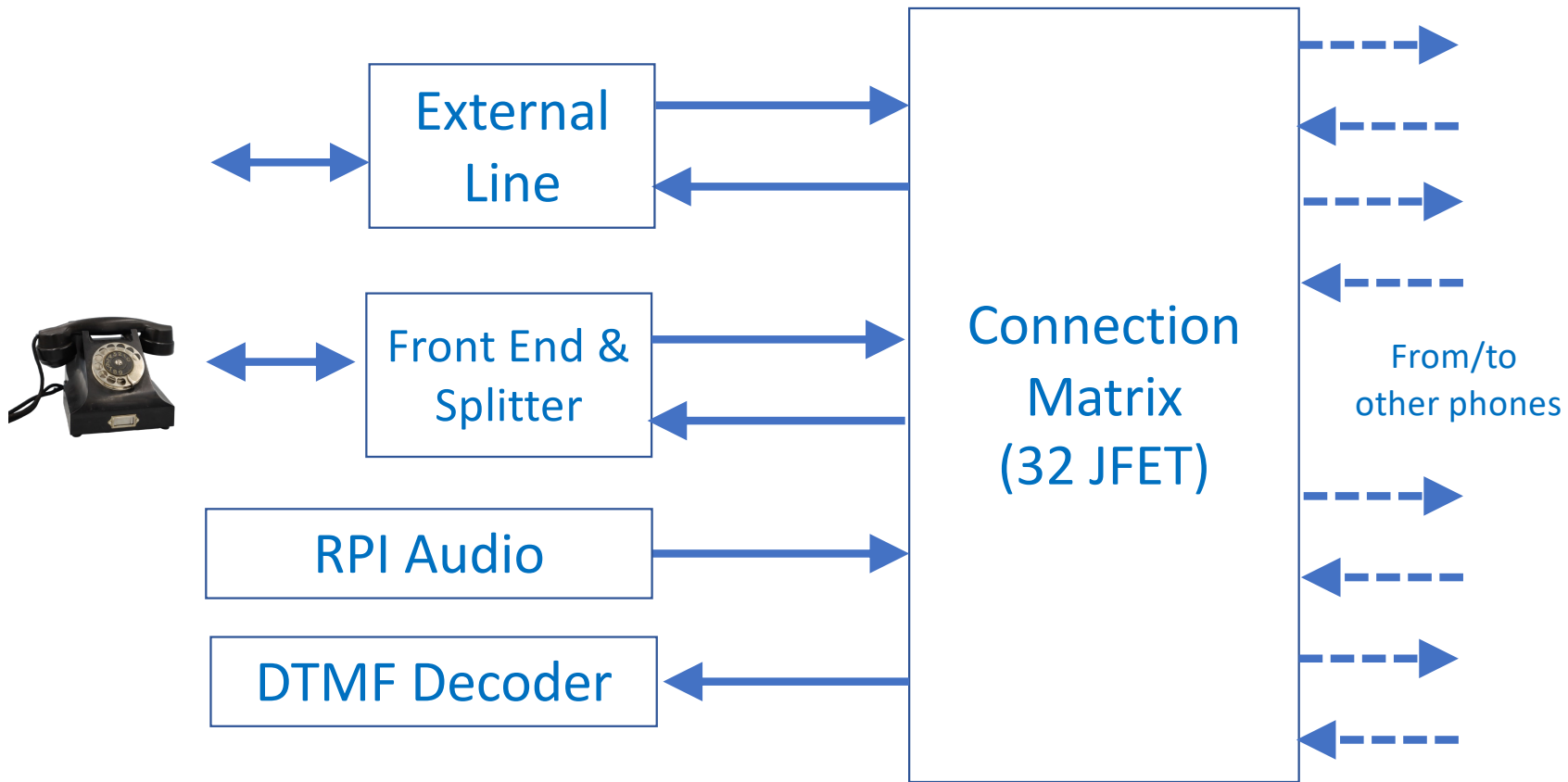|	Dial Tone/Hz	Engaged/Hz	Ringing/Hz	Unobtainable/Hz
Germany	425	425	425	950-1400-1800
England	350+450	400	400+450	400
US	350+440	440	440+480	Message

Possible Improvements

- Use IC Matrix to be able to connect multiple parties
- Expand Main-FSM for multiple parties
- Use cheaper Raspberry-Pi (RPI-Nano)
- Program with Micro-Python
- Improve Splitter
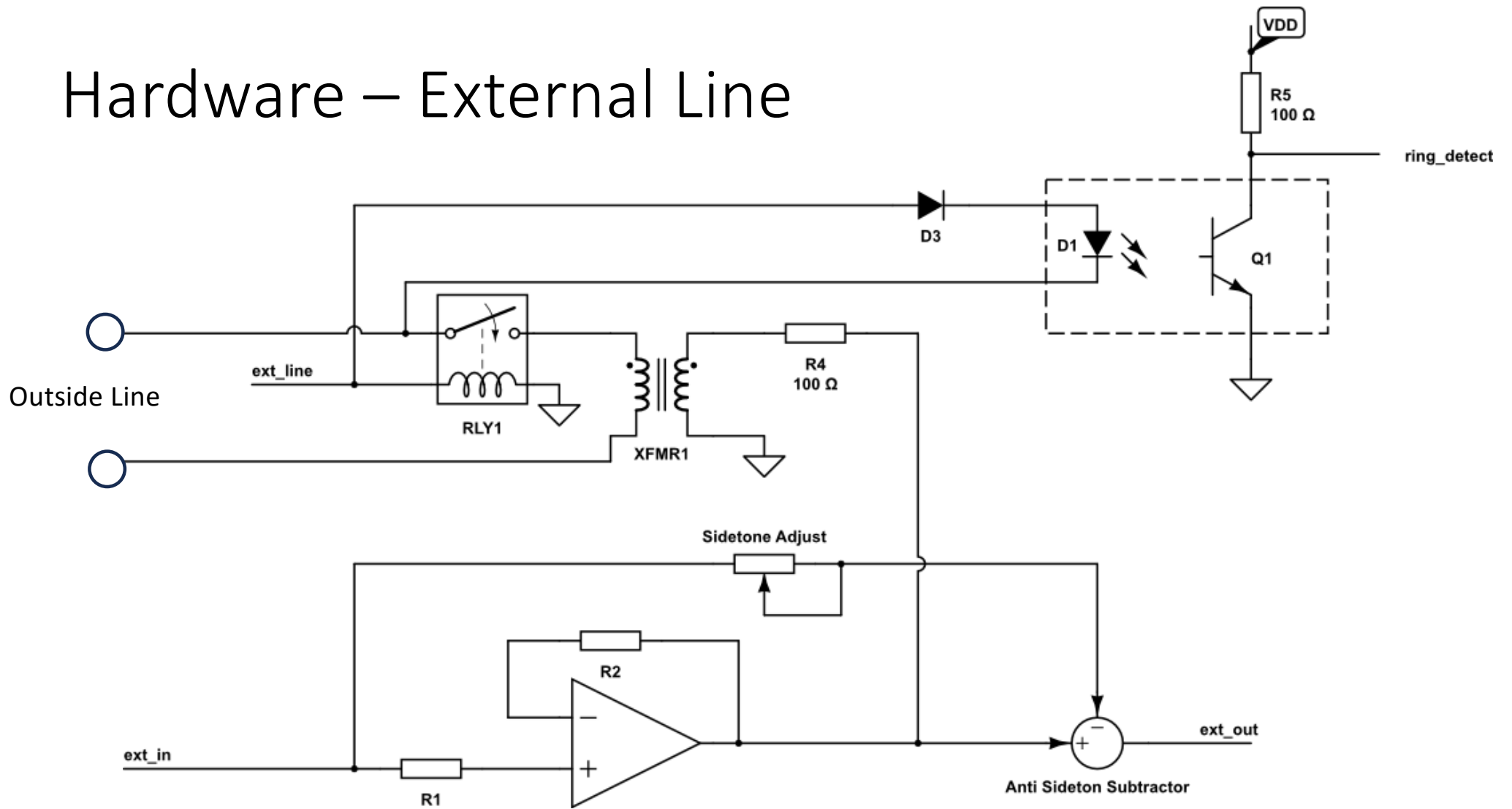- Less Power Consumption (One processor, lower clock frequency)

`https://github.com/hansgelke/retro_v3`

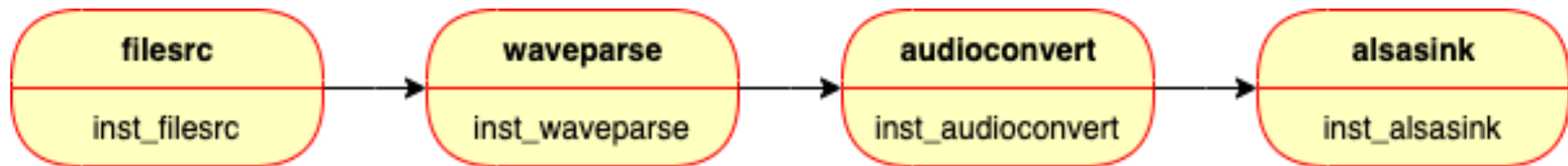**Thanks for listening**

Hardware – Connection Matrix

External Line

Front End & Splitter

RPI Audio

DTMF Decoder

Connection Matrix (32 JFET)
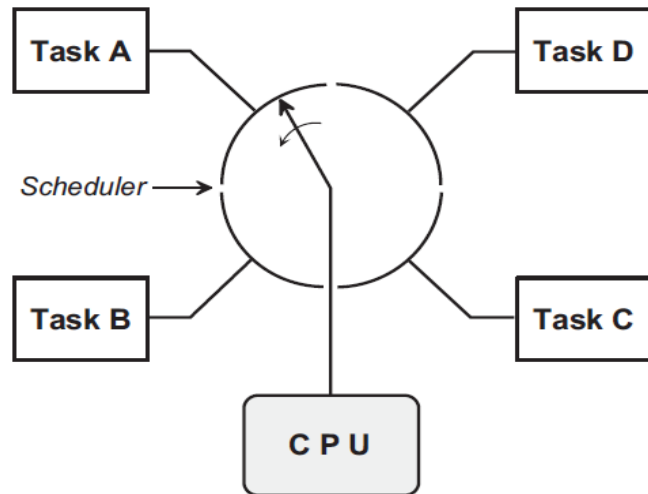
From/to other phones
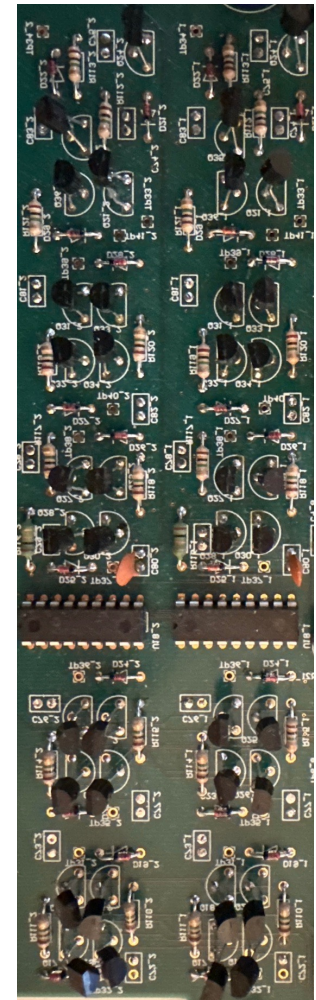
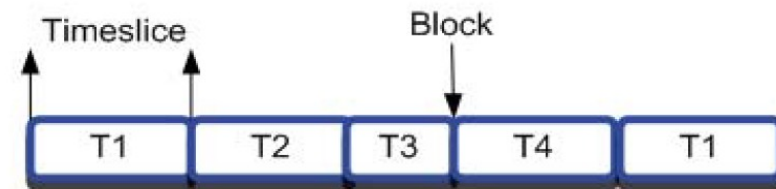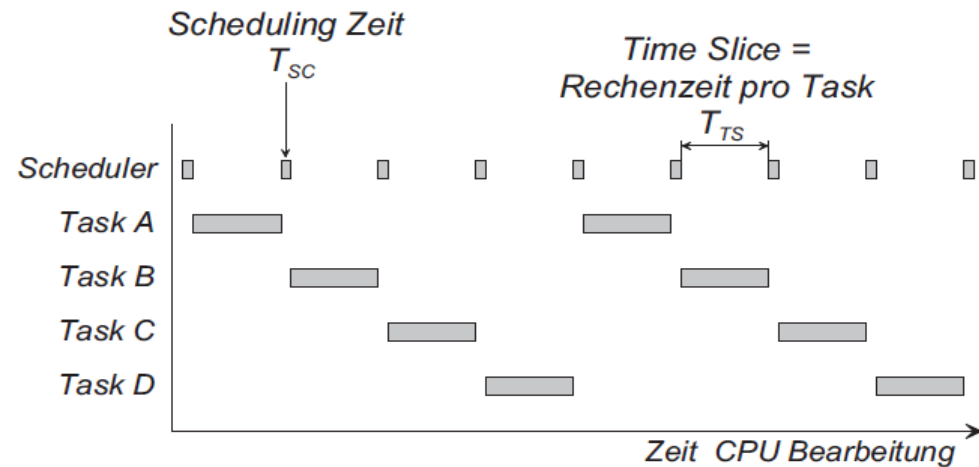# Hardware – External Line

Software

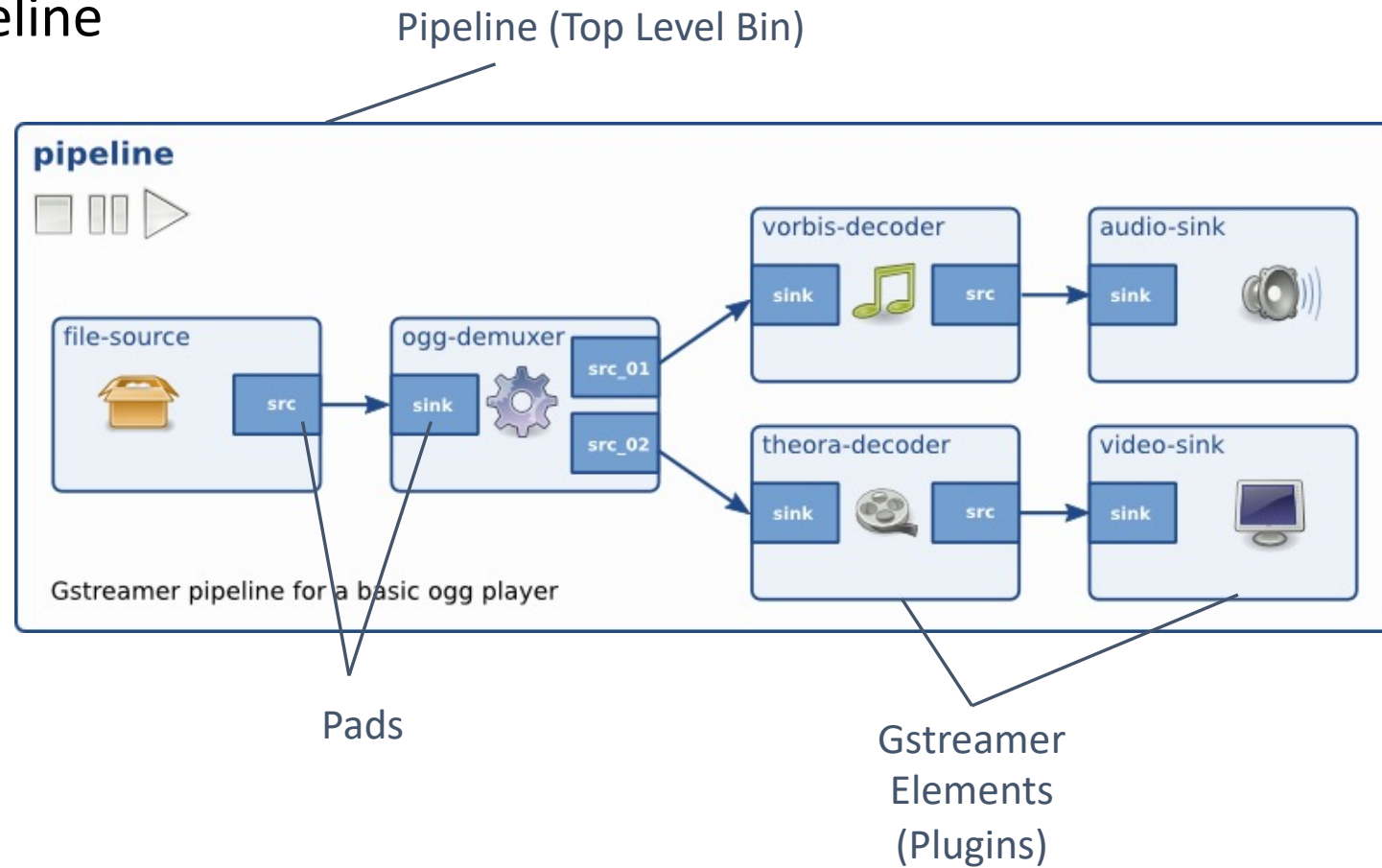- GST Pipeline for Announcements

Software

- Realtime Scheduling

Example: Round Robbin Timeslicing

Software

- GST Pipeline

Pipeline (Top Level Bin)

Pads

Gstreamer
Elements
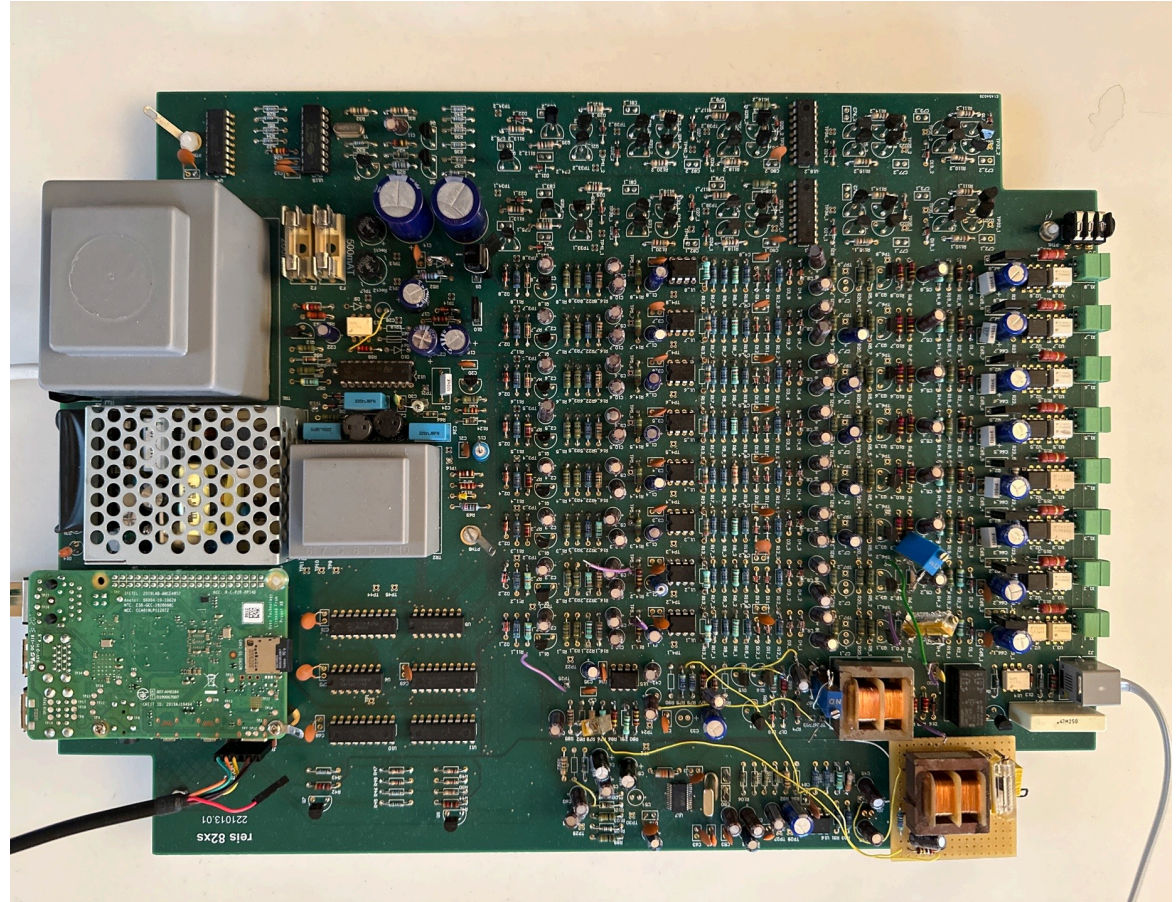(Plugins)

Features

- Central switch for eight phones
- External calls via analog phone line
- Looks and feel of phones world wide (implemented: German, US, UK)
- C-Software for Yocto Linux running on Raspberry Pi 4 (RPI)

Splitter

- Example Audio Splitting