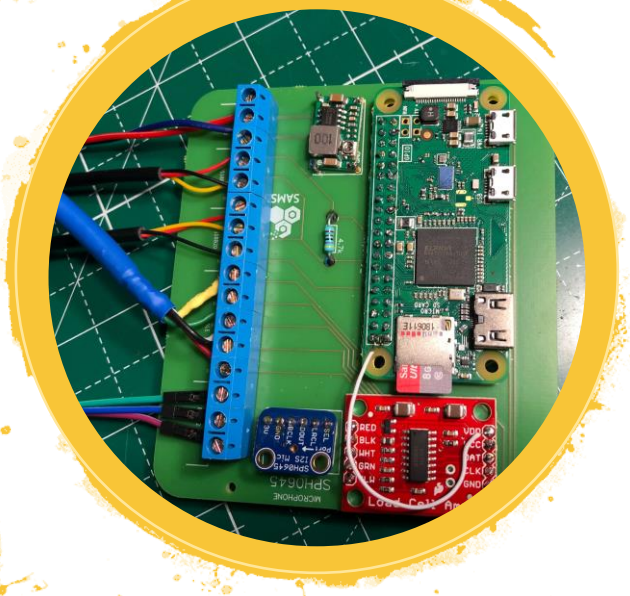


Beehive Monitoring System

Sascha Fiedler

University of Kassel, Germany
Organic Agricultural Sciences
Agricultural and Biosystems Engineering

Witzenhausen, 9. January 2019

AGRARTECHNIK WITZENHAUSEN

UNIKASSEL VERSITÄT | ORGANIC AGRICULTURAL SCIENCES

Smart Apiculture Management Services

SAMS

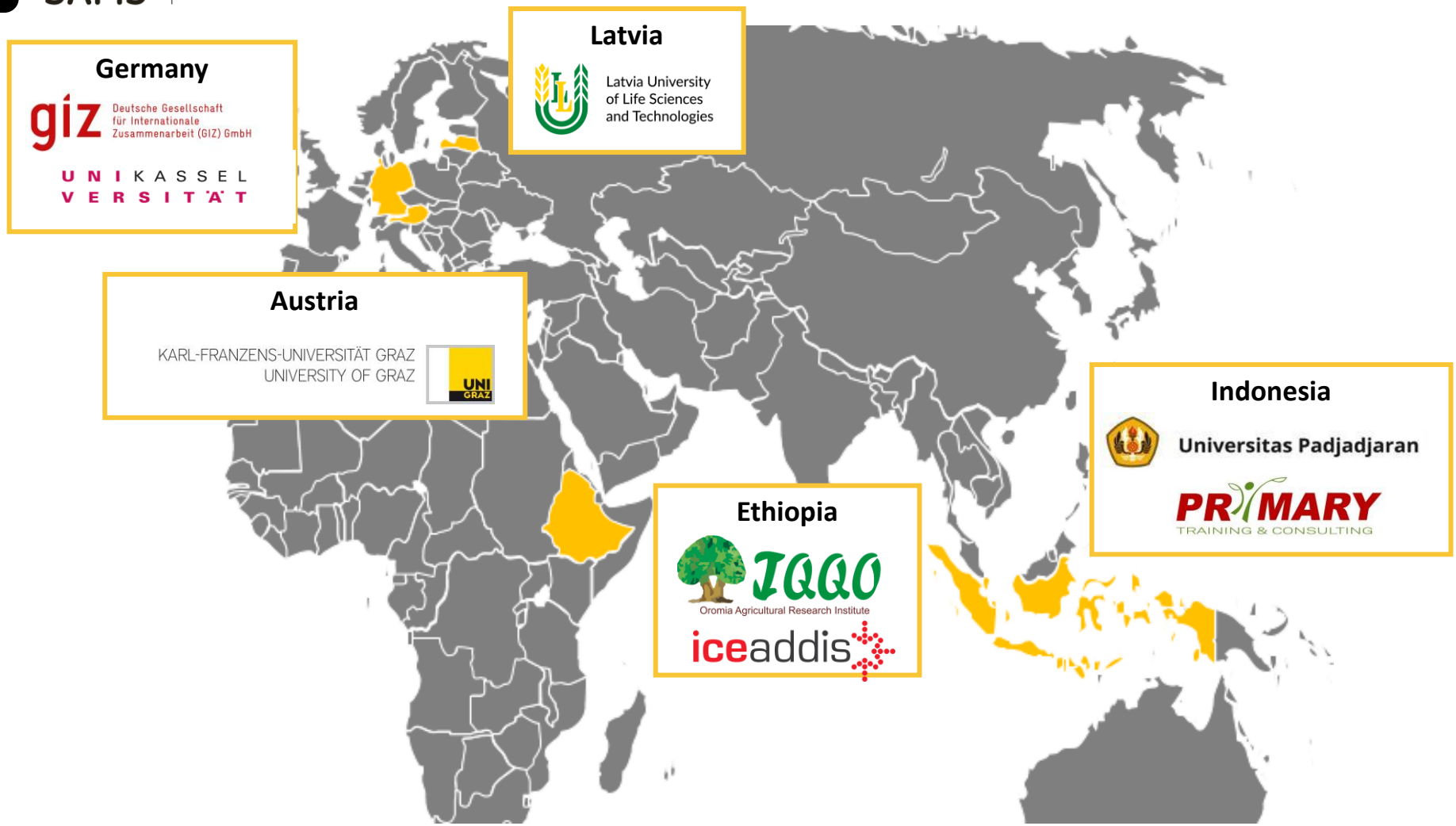
This project has received funding from the European Union´s Horizon 2020 research and innovation programme under Grant Agreement N° 780755. The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the EU

International Partnership

SAMS – Smart Apiculture Management Services

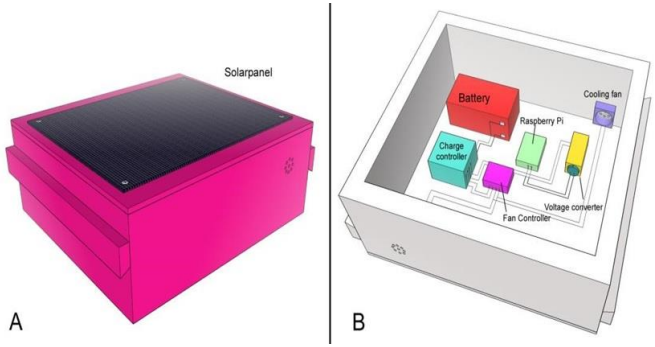
SAMS Smart Apiculture Management Services

The SAMS consortium
8 partners - 5 countries - 3 continents



SAMS Smart Apiculture Management Services

SAMS in a nutshell

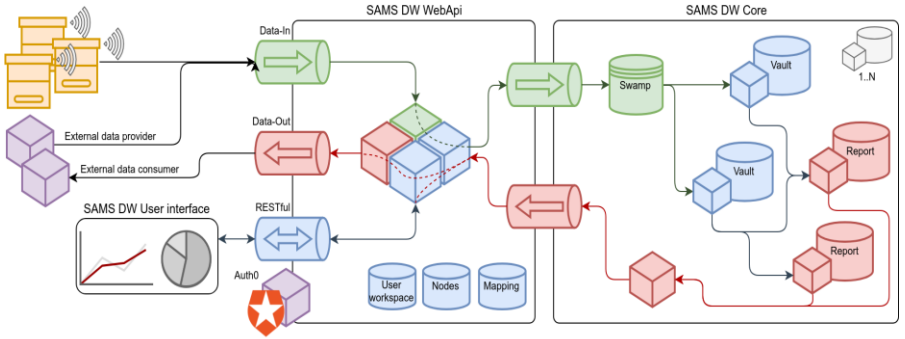


SAMS supports International Partnership Building in low and middle income countries in ASEAN and sub-Saharan Africa

Objective	Strengthen the international cooperation of the EU with developing countries in ICT concentrating the field of sustainable agriculture as a vehicle for rural areas.
Duration	01/2018 – 12/2020
Budget	EUR 1.99 million
Funded by	European Union's Horizon 2020 Research and Innovation Programme

SAMS as an ICT solution …

- allows active monitoring and managing of bee colonies
- ensures bee health and bee productivity
- gives answers to the requirements of beekeeping in developing countries
- is available as an open source technology

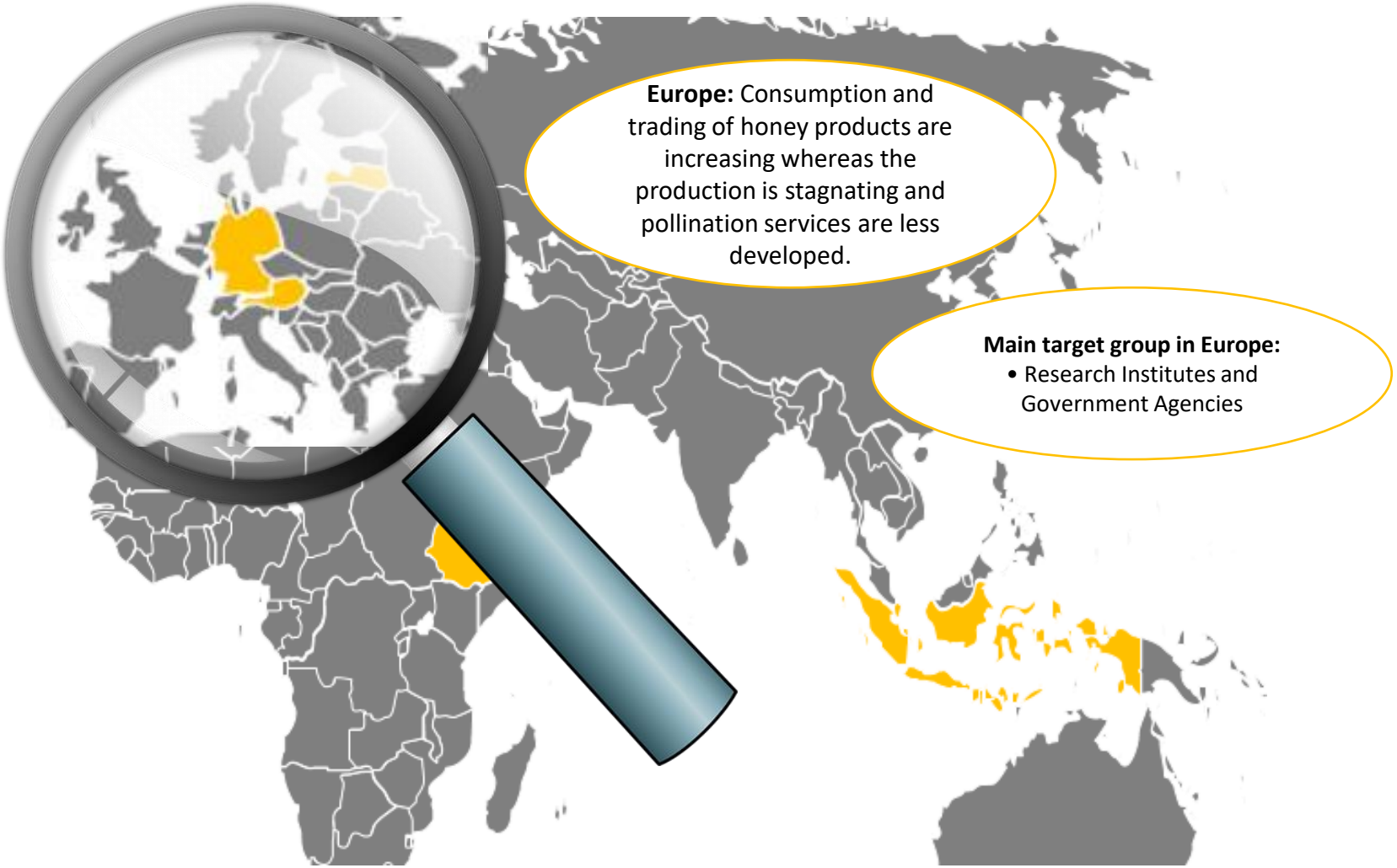


Bees play a key role in the preservation of our ecosystem, the global fight against hunger and in ensuring our existence.

Smart Apiculture Management Services

SAMS

Why SAMS?
three continents – three different scenarios

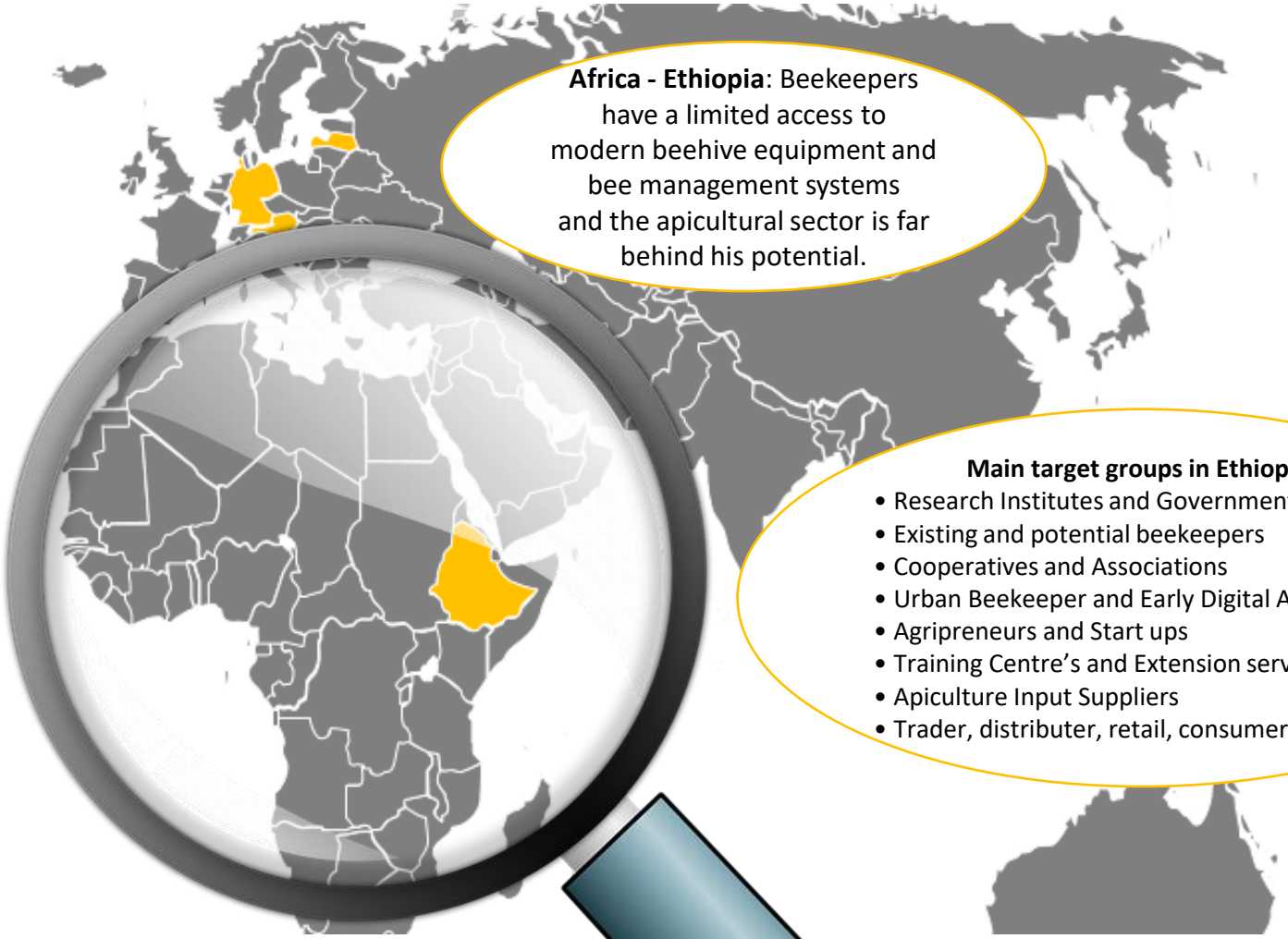


Europe: Consumption and trading of honey products are increasing whereas the production is stagnating and pollination services are less developed.

Main target group in Europe:
- Research Institutes and Government Agencies

SAMS Smart Apiculture Management Services

Why SAMS?
three continents – three different scenarios



Africa - Ethiopia: Beekeepers have a limited access to modern beehive equipment and bee management systems and the apicultural sector is far behind his potential.

Main target groups in Ethiopia:

- Research Institutes and Government Agencies
- Existing and potential beekeepers
- Cooperatives and Associations
- Urban Beekeeper and Early Digital Adopters
- Agripreneurs and Start ups
- Training Centre's and Extension services
- Apiculture Input Suppliers
- Trader, distributer, retail, consumer

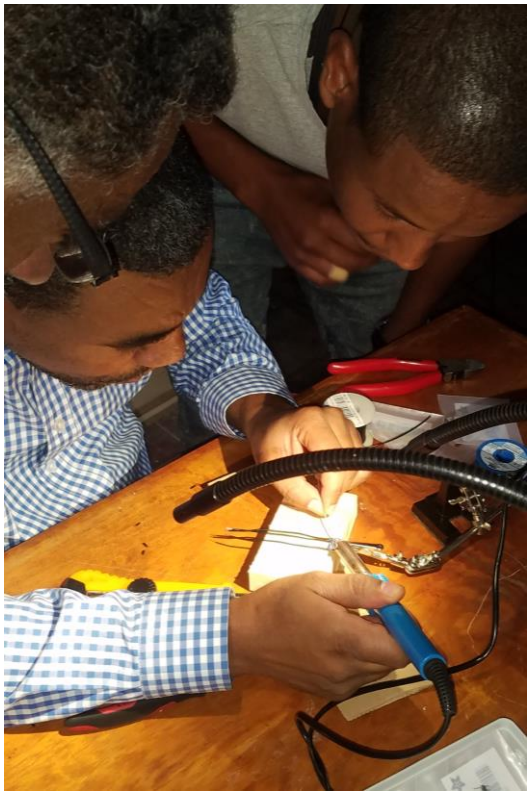
Smart Apiculture Management Services

SAMS

Installation Workshop Ethiopia

SAMS

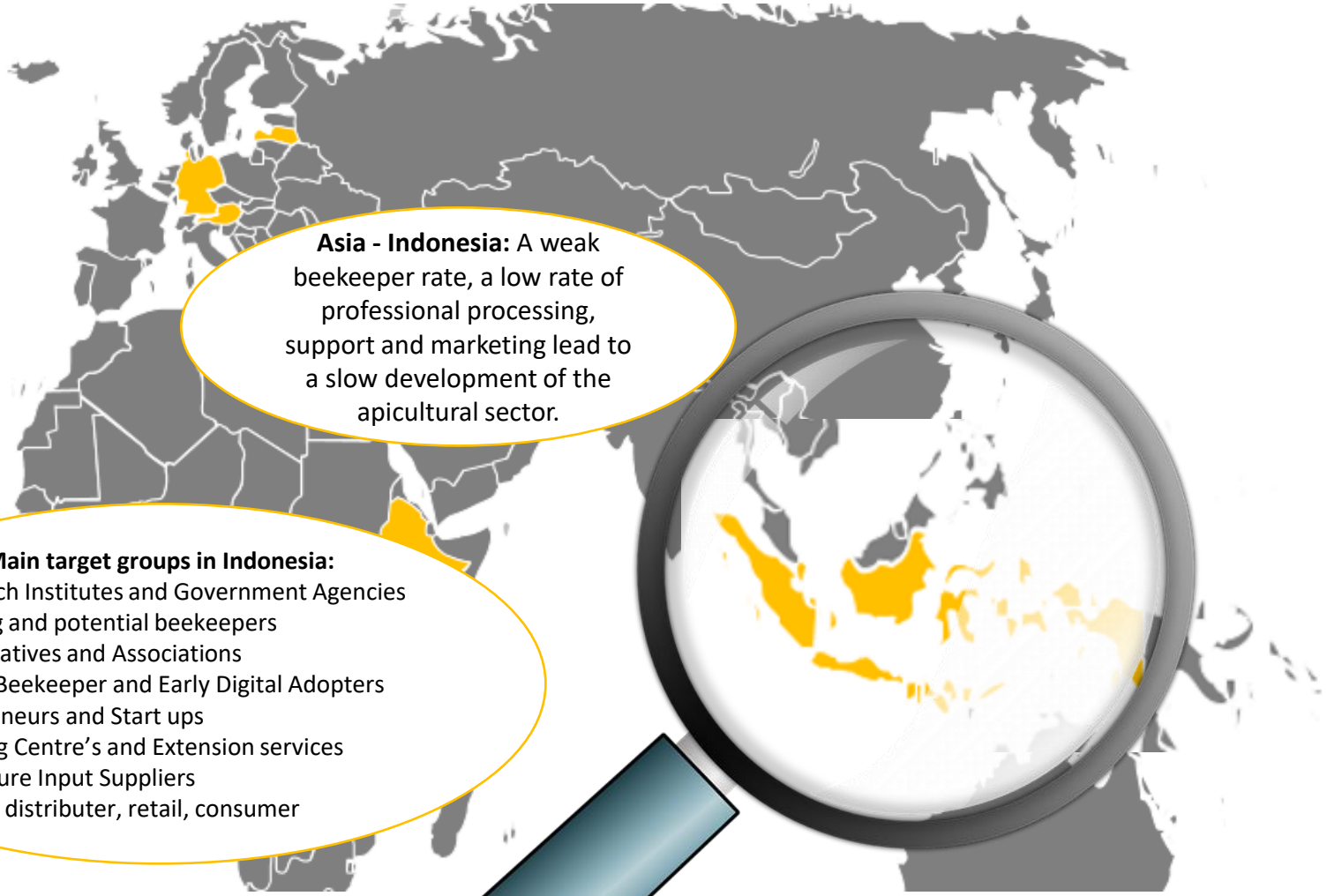
Smart
Apiculture
Management
Services



Smart Apiculture Management Services

SAMS

Why SAMS?
three continents – three different scenarios



Asia - Indonesia: A weak beekeeper rate, a low rate of professional processing, support and marketing lead to a slow development of the apicultural sector.

Main target groups in Indonesia:

- Research Institutes and Government Agencies
- Existing and potential beekeepers
- Cooperatives and Associations
- Urban Beekeeper and Early Digital Adopters
- Agripreneurs and Start ups
- Training Centre's and Extension services
- Apiculture Input Suppliers
- Trader, distributer, retail, consumer

Smart Apiculture Management Services

SAMS

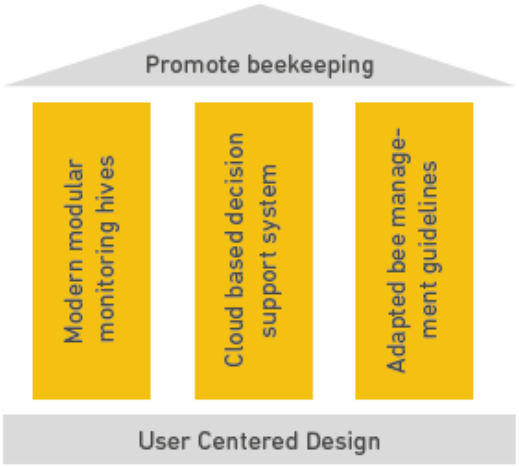
Installation Workshop Indonesia


AQUA
BARON
"OOO"
BABAD

SAMS Smart Apiculture Management Services

SAMS Smart Apiculture Management Services

Implementation actions of SAMS



1st Identify preconditions, potentials and user´s needs through research and context analysis in SAMS partner countries

2nd Include end-user in solution development of the (1) Advisory and (2) Decision Support System through SAMS User Centered Design Cycles

3rd Set up of first pilot proto-hives for further iterations and adaptions of the developed ICT solution to regional settings and users needs

4th Enable the end-user to monitor and manage bee colonies in a sustainable and efficient way by taking also bee-health aspects in consideration

5th Development of business concepts along the honey value chain

Smart Apiculture Management Services

SAMS

Objectives of SAMS

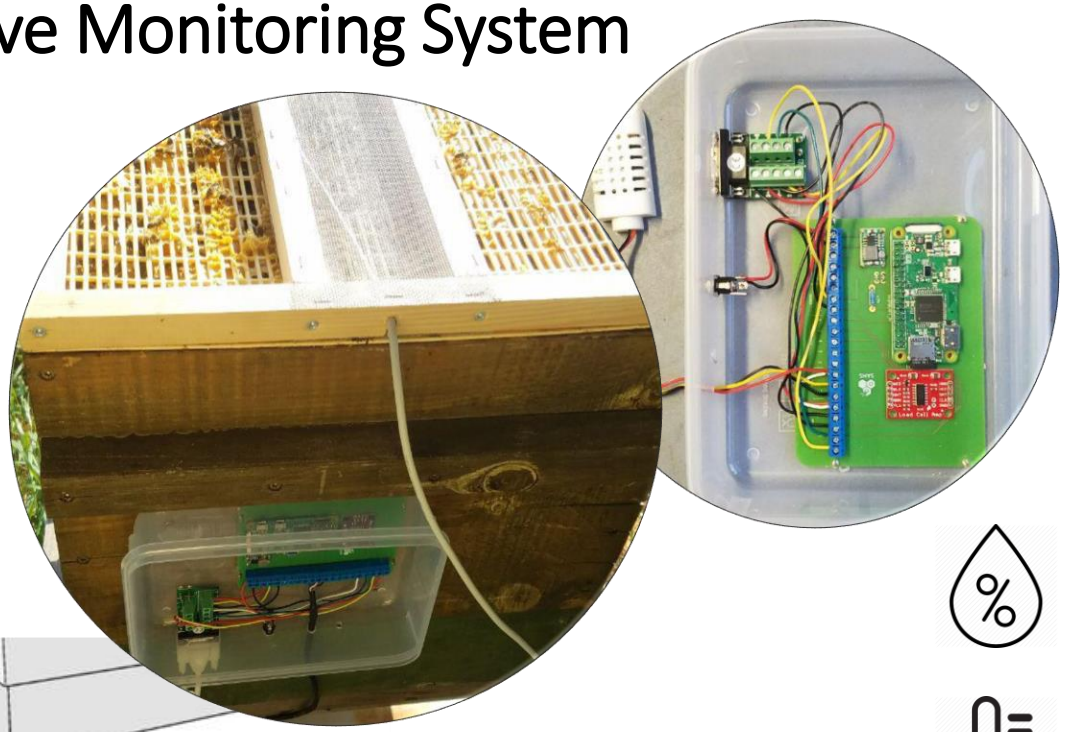
- ✓ Strengthen international cooperation of the EU with developing countries in ICT
- ✓ Address requirements of communities and stakeholders to promote and advance forms of existing beekeeping
- ✓ Monitor bee colonies in Germany, Ethiopia and Indonesia through an open source technology
- ✓ Evaluate gained information and convert them into recommendations for beekeepers
- ✓ Overcome country-specific challenges of beekeeping and simplify the management of bee colonies
- ✓ Strengthen interregional and international bee related partnership and cooperation
- ✓ Gain information on bee mortality

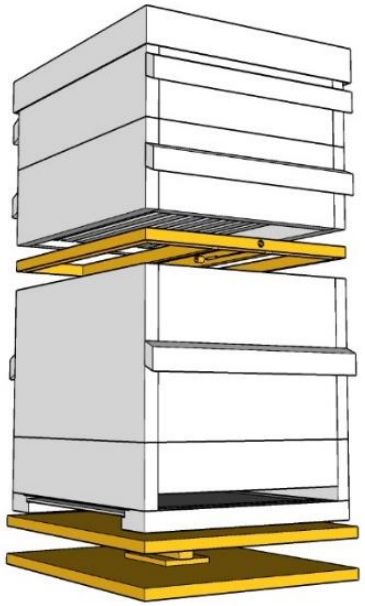
Smart Apiculture Management Services

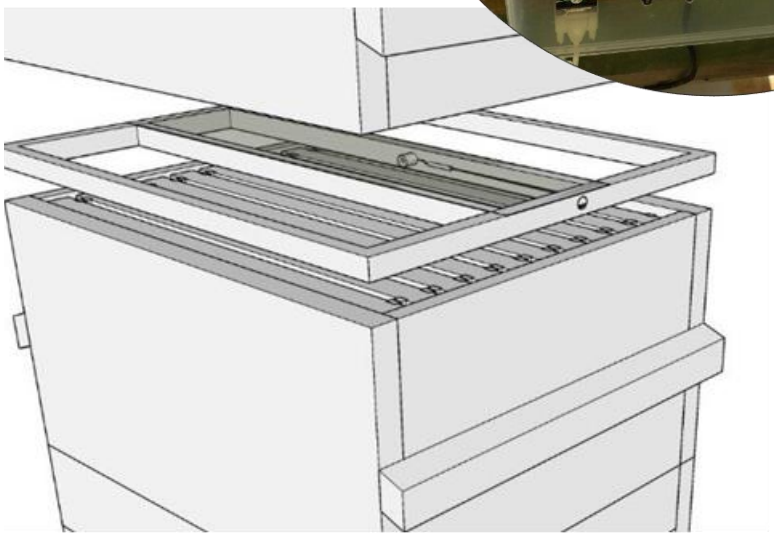
SAMS

Beehive Monitoring System

Remote Sensing Technology
For Active Monitoring
On Bee Health And Productivity

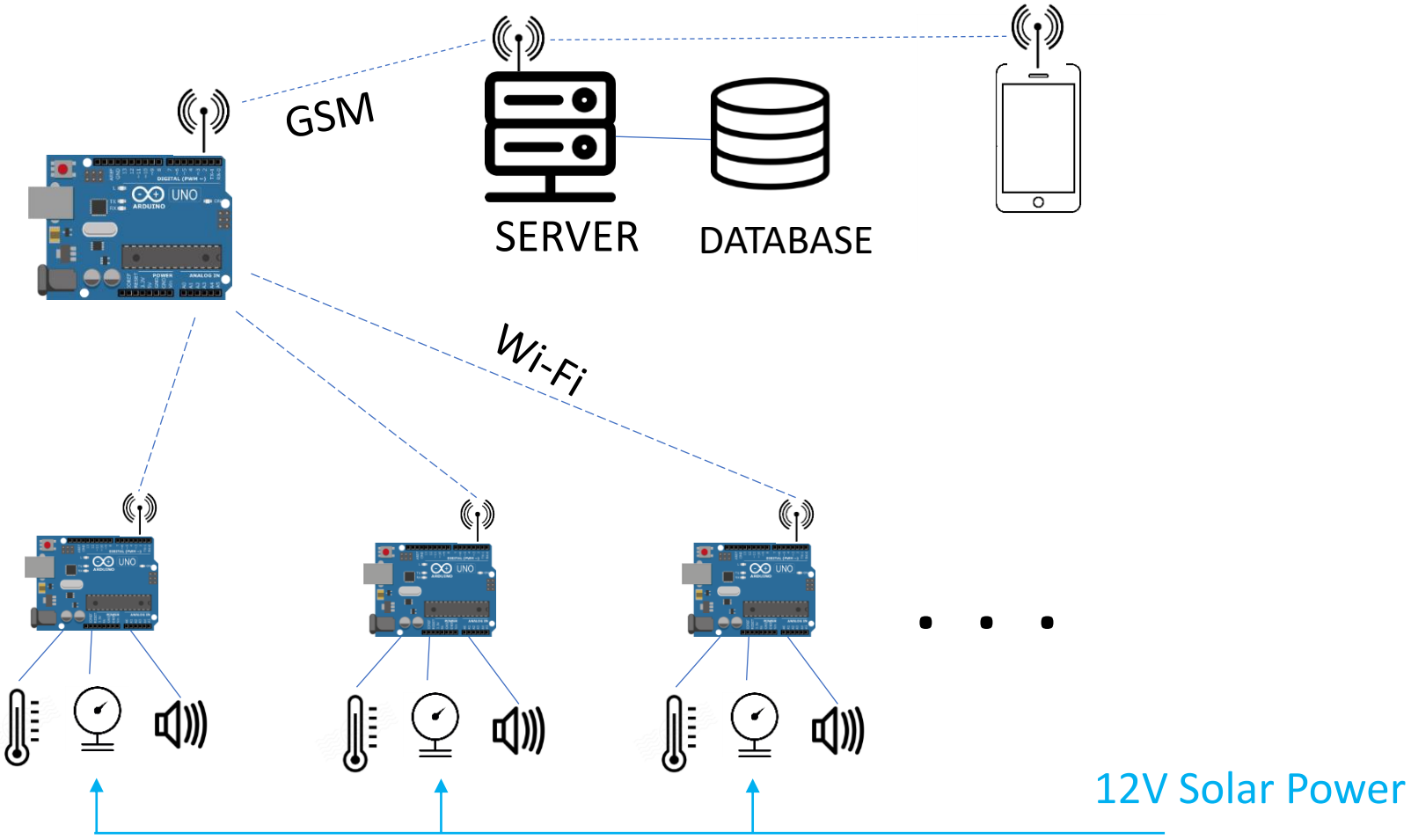






SAMS Smart Apiculture Management Services

Monitoring System Structure



Benefits of bee colony remote monitoring

- Decrease of management costs
- Decrease the number of on-site inspections
- Less disturbance to bees
- Decrease the burden of death rate
- Increase of bee colony production

- Remote monitoring main goal is to identify **different states** of the bee colony and **prevent colony losses!**

Smart Apiculture Management Services

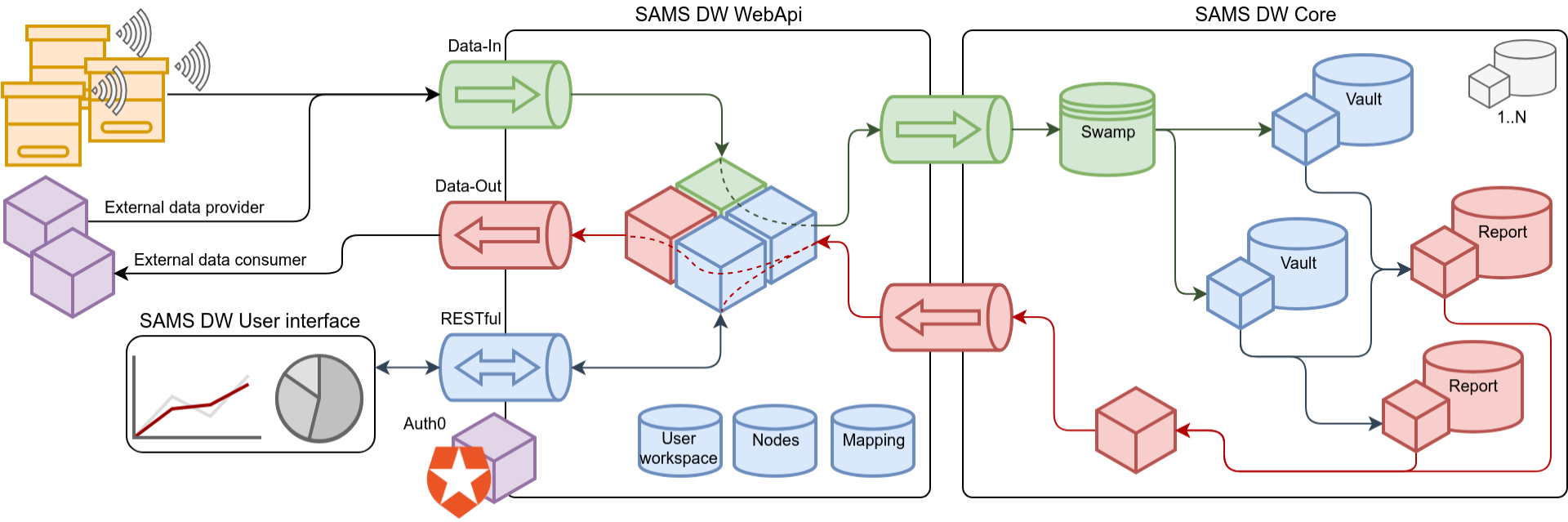
Bee colony states

- Death
- Pre swarming / Swarming
- Brood rearing
- Broodless state
- Queenless state
- Starvation
- Nectar flow

Smart Apiculture Management Services

SAMS

Data Storage and and Processing: SAMS Data Warehouse



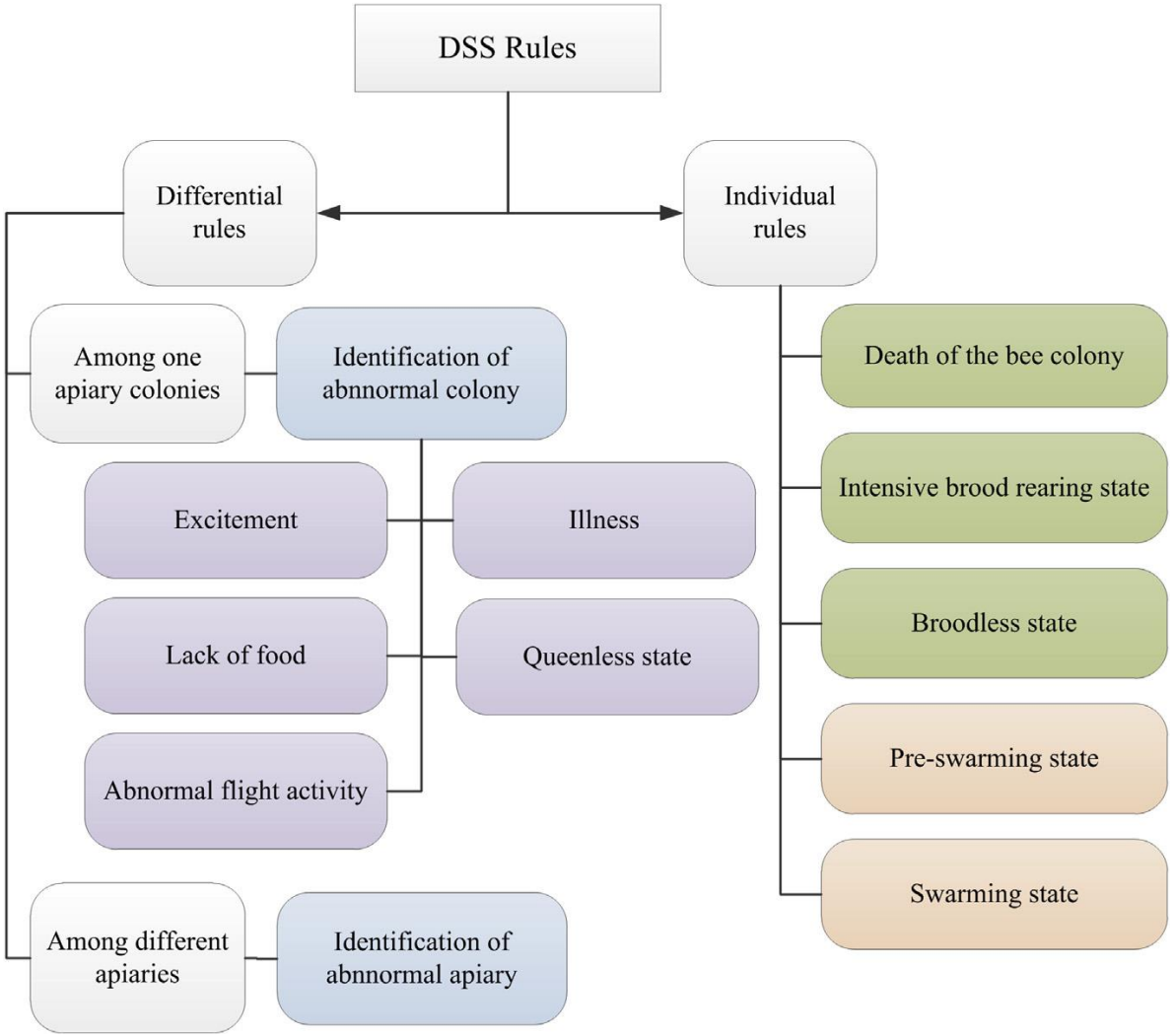
Decision Support System (DSS)

- In general: «any system which helps someone with any aspect of making a decision» (M. Bruen)
- Decision – a choice between options («do nothing» is also included)
- DSS main characteristics regarding the user:
  - supports decision makers rather than replaces them;
  - is under control of the DSS user;
  - is interactive and user-friendly;

Smart Apiculture Management Services

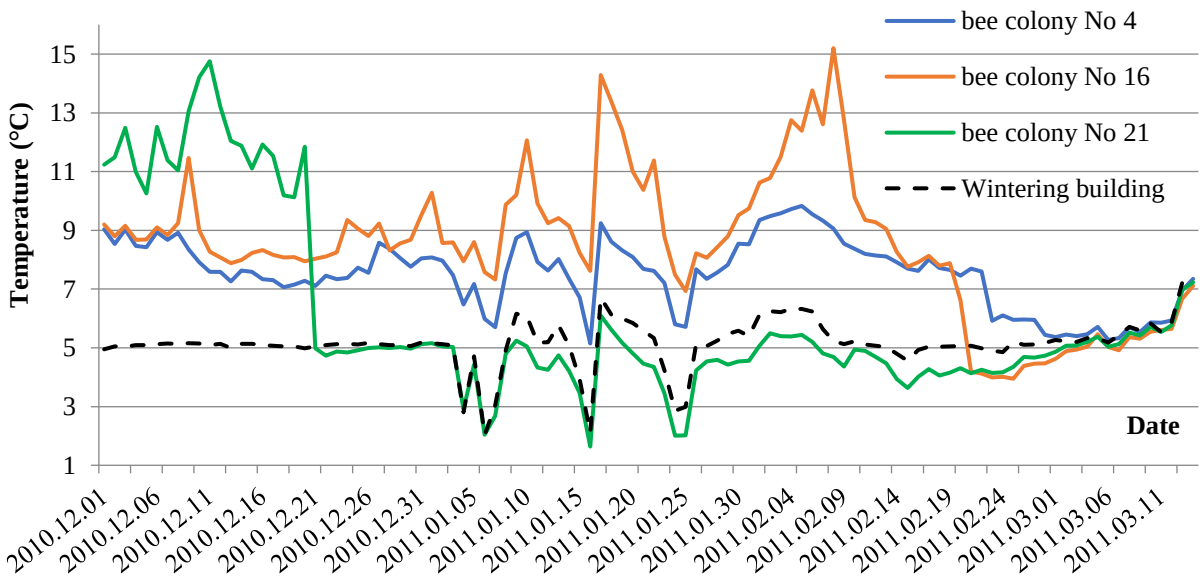
SAMS

DSS rules



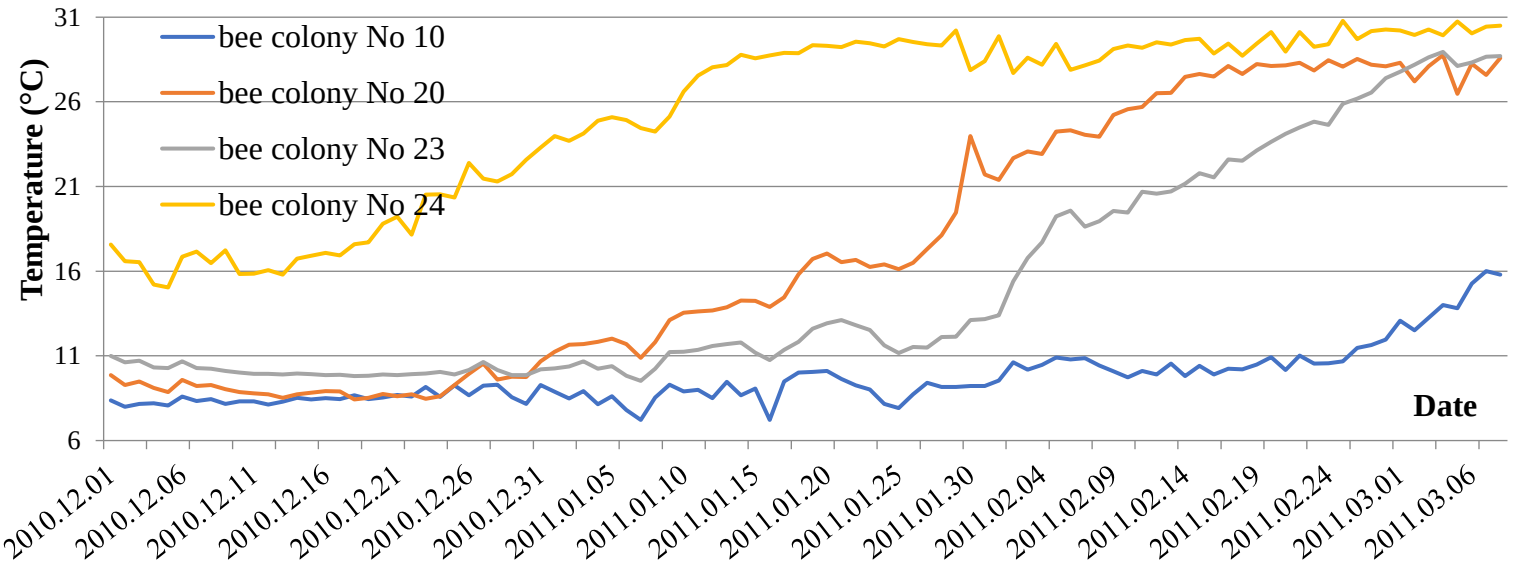
DSS rule: Colony death

- The developed algorithm is based on the last temperature data comparison between the temperature inside the hive and the ambient temperature



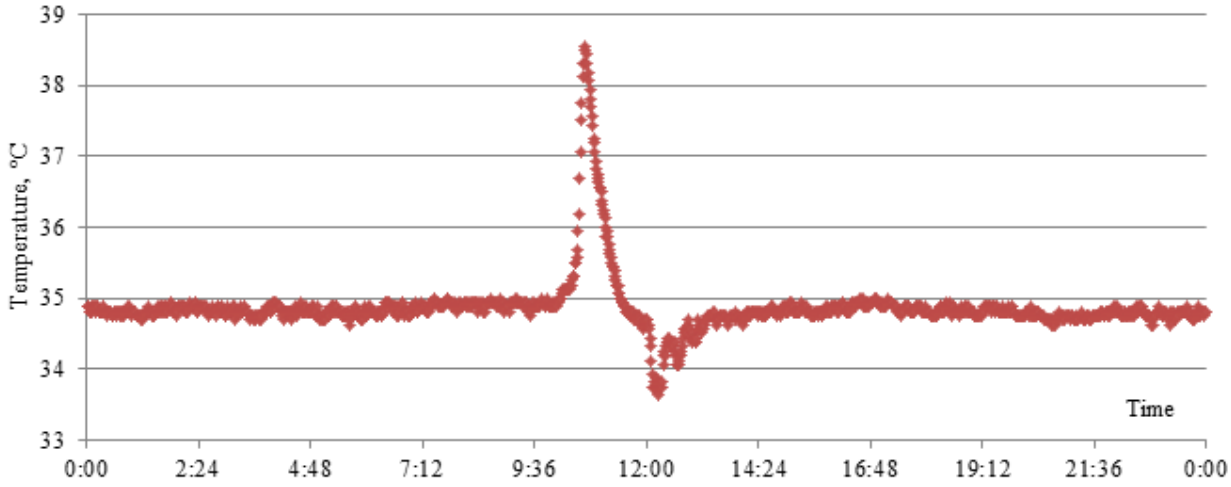
DSS rule: Brood rearing

- Brooding is characterized by significant temperature growth (inside the colony cluster, the temperature reaches approx. +35 ºC)



DSS rule: Swarming state

- Swarming – queen with a large number of worker bees leave the hive to establish a new nest
- Detection of swarming event using a neural network approach

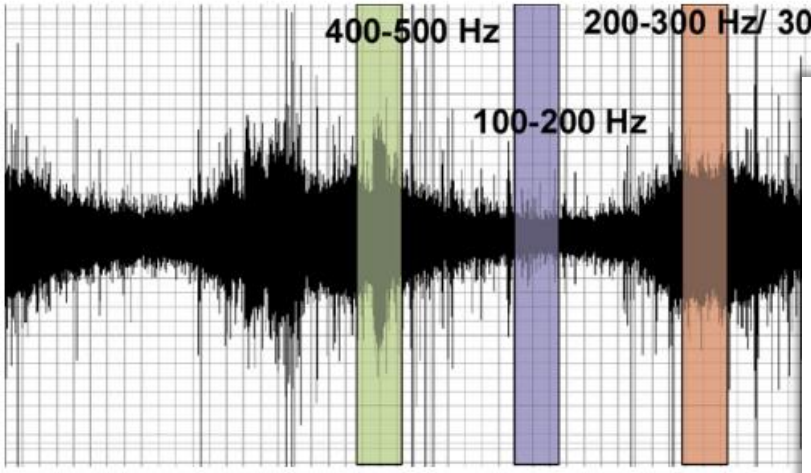


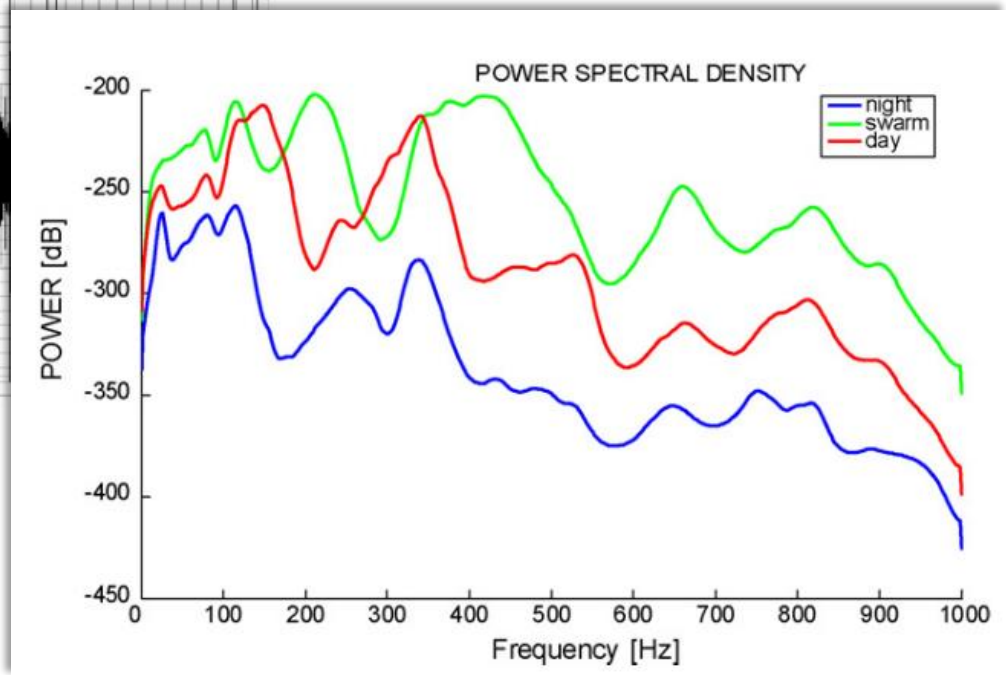
Weight monitoring

- The occurrence of nectar flow during the foraging season (start and end of nectar flow) or daily gain in nectar stores
- Consumption of food during non-foraging periods
- The occurrence of swarming events through a decrease in the hive weight

Soundanalysis

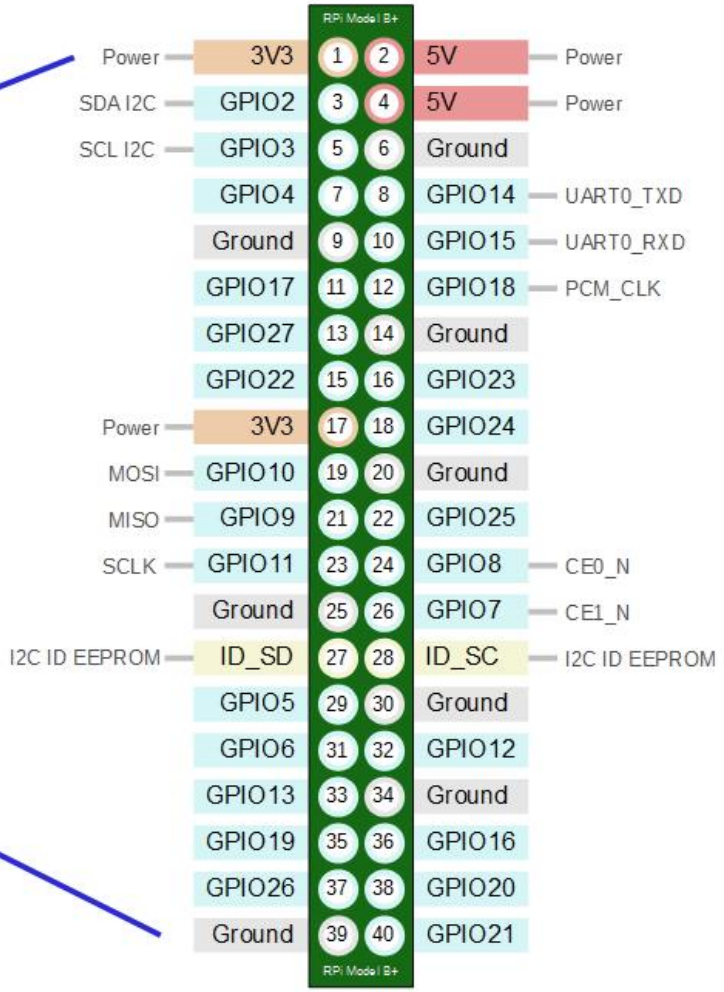
Fast Fourier Transformation (FFT): From noise to spectral function





SAMS Smart Apiculture Management Services

RaspberyPi Zero W: GPIO Pinout Orientation

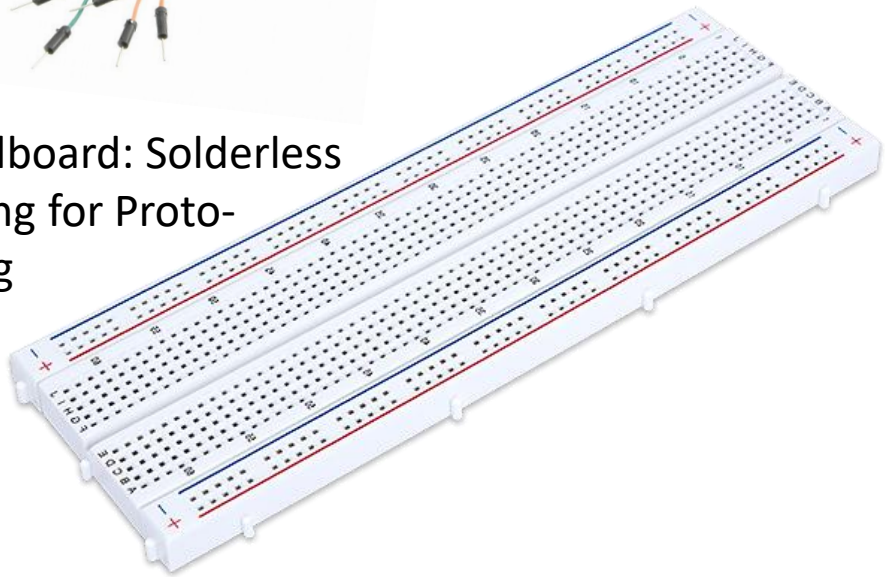


Smart Apiculture Management Services

SAMS

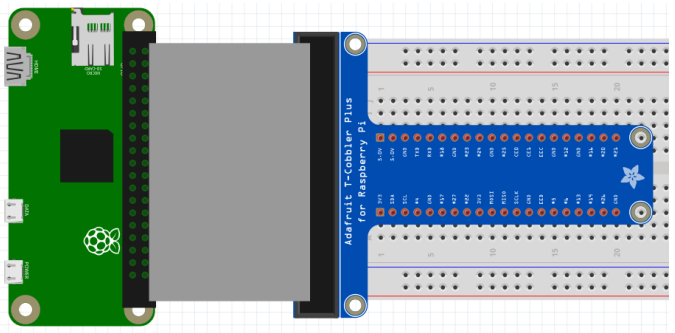
Breadboard and Tools

Solderless Breadboard Jumper Cable Wires



Breadboard: Solderless wireing for Proto-typing

T-Cobbler: GPIO Breakout Adapter for Raspberry Pi



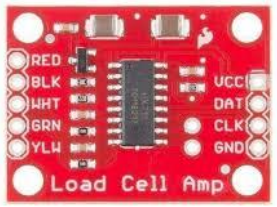
SAMS Smart Apiculture Management Services

Adding Components to your Raspberry Pi

Power Management + RTC:
UUGear WittyPi mini+ DS3231SN

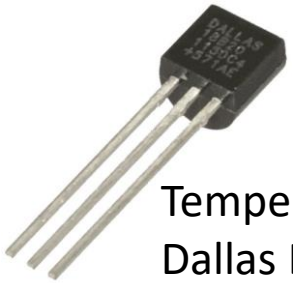
Load cell:
BOSCHE H30A

Computer:
Raspberry Pi Zero W



A/D Converter:
AVIA HX711

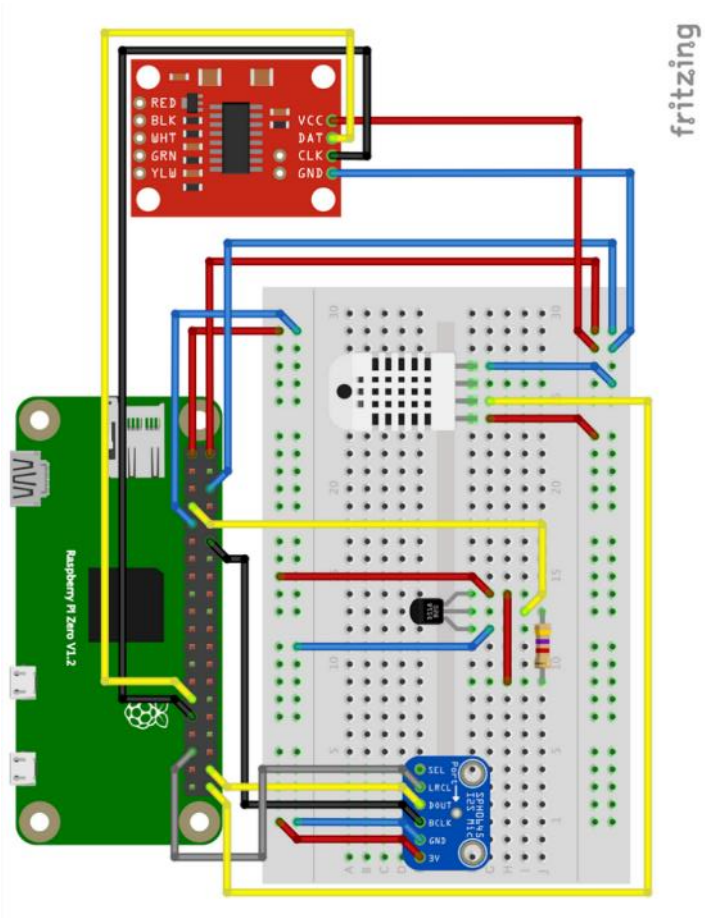
Humidity:
Aosong DHT22



Temperature:
Dallas DS18B20

Microphone:
Adafruit I2S MEMS
SP0645

GPIO Wirering: Wire Components with your Pi

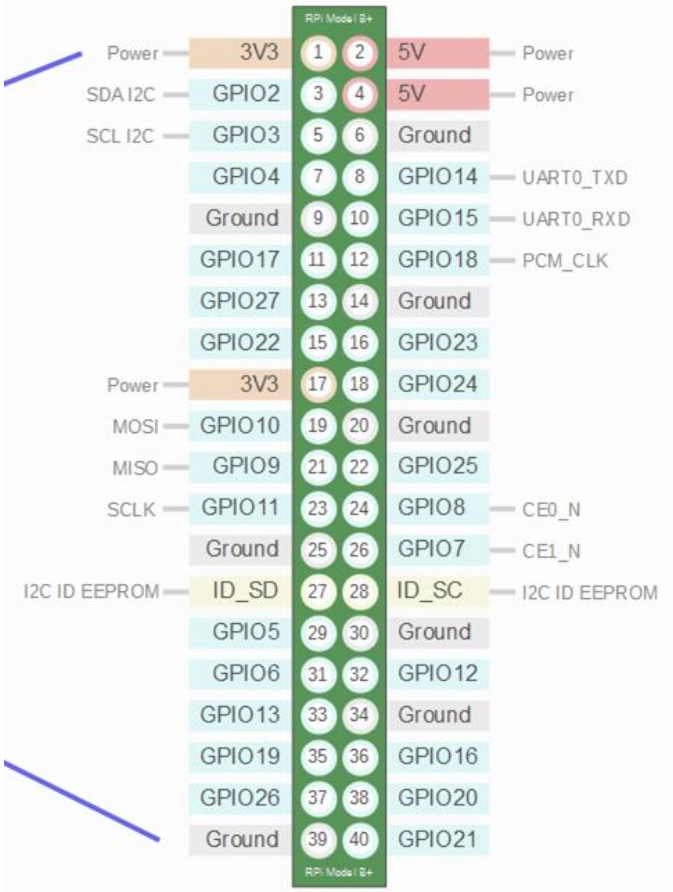


<u>A/D Converter HX711</u>
VCC to 5V
DAT to Pin29/ GPIO 5
CLK to Pin31/ GPIO 6
GND to GND

<u>Microphone SPH0645</u>
LRCL to Pin35/ GPIO 19
DOUT to Pin38/ GPIO 20
BCLK to Pin12/ GPIO 18
GND to GND
3V to 3V

<u>Temperature DS18B20</u>
VDD to 3V
DAT to Pin13/ GPIO 27
GND to GND
(4,7kΩ VDD und DAT)

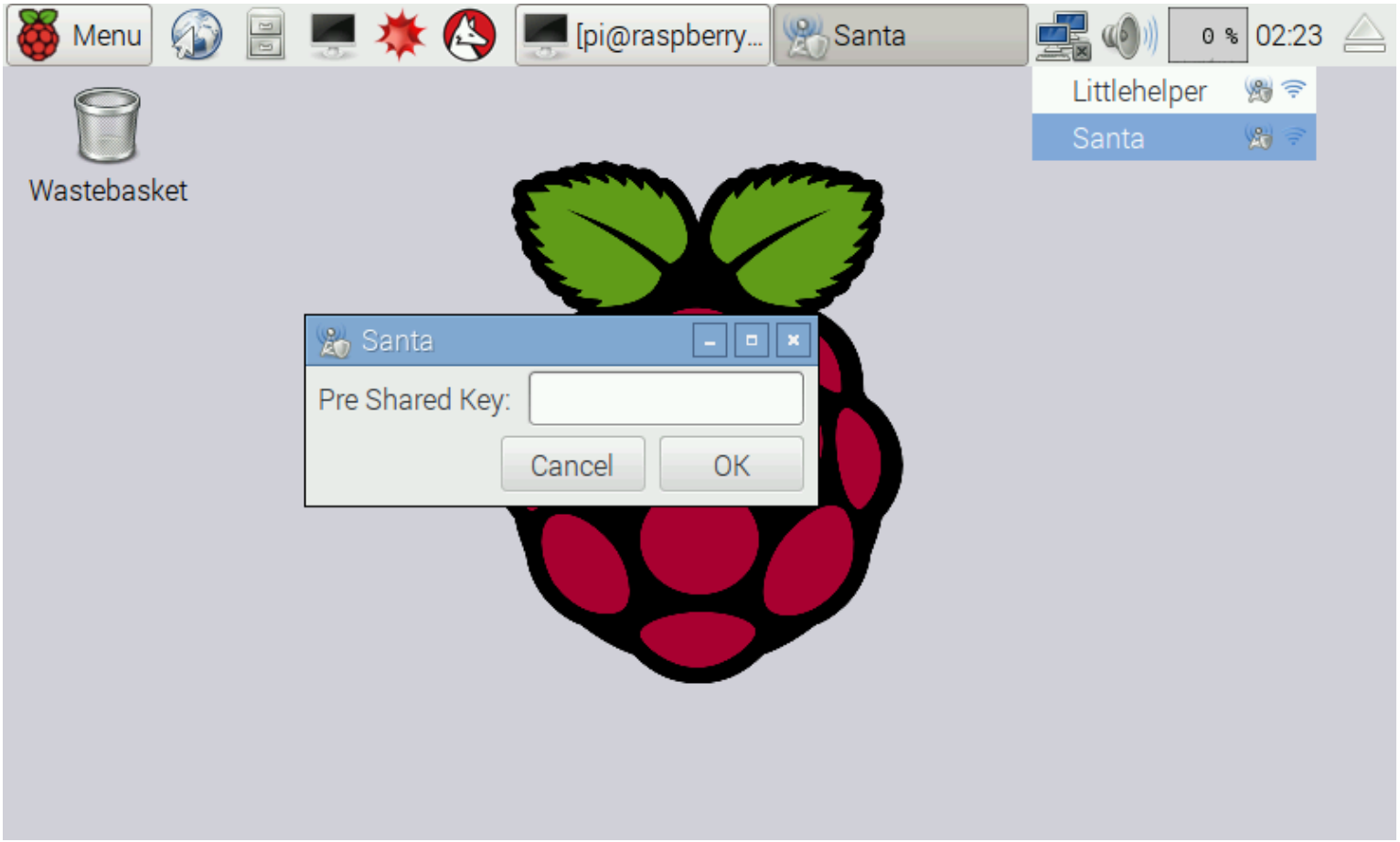
<u>Humidity DHT22</u>
VDD to 5V
DAT to Pin40/ GPIO 21
GND to GND



Smart Apiculture Management Services

SAMS

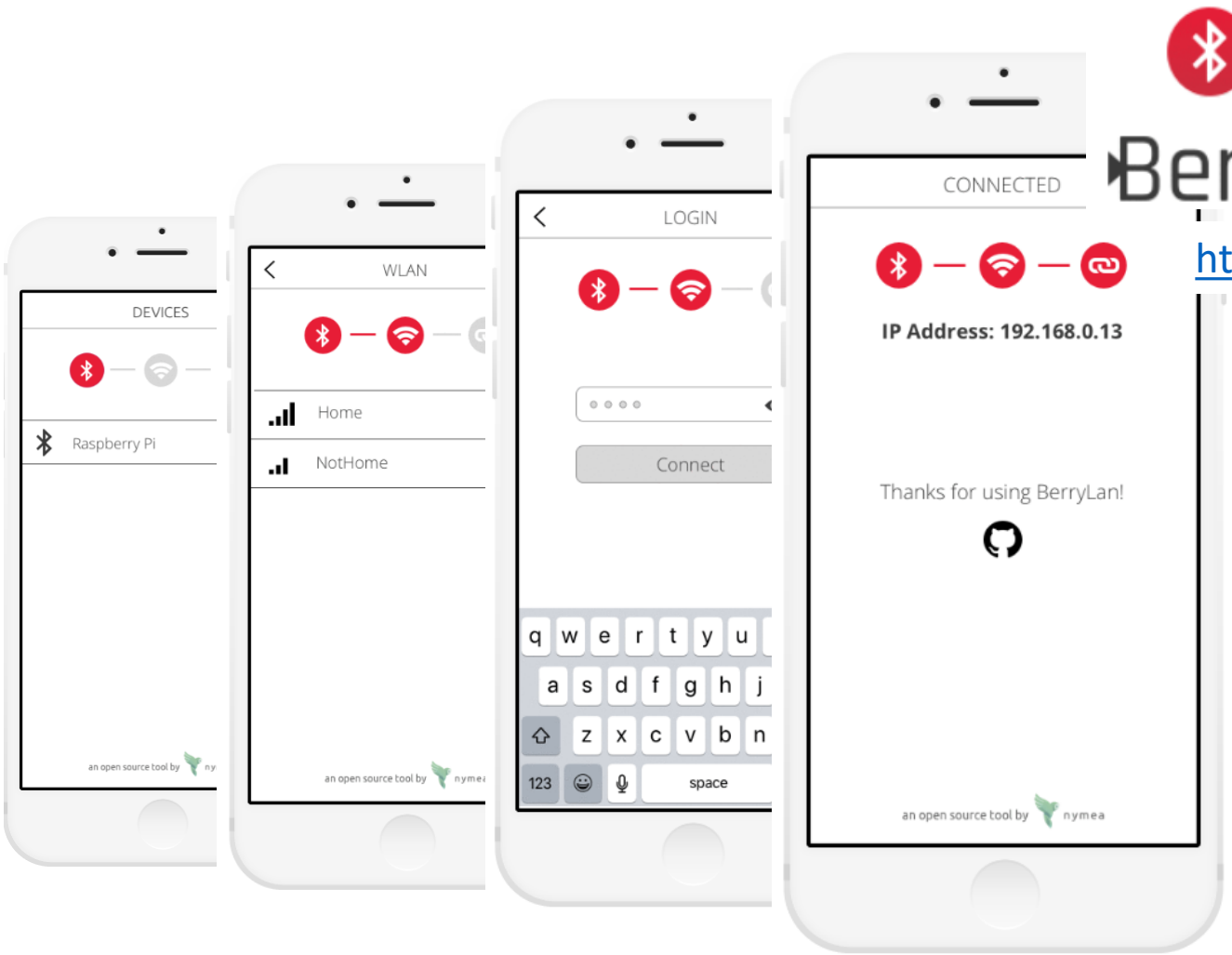
Set up WiFi on your Pi with GUI



Smart Apiculture Management Services

SAMS

Set up WiFi on your Pi with BerryLan



BerryLan

http://www.berrylan.org/

Set up WiFi on your Pi via wpa_supplicant File

WPA Supplicant File with WiFi SSID

Create *wpa_supplicant.conf* file in boot folder on SD Card:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=<Insert country code here>

network={
 ssid="<Name of your WiFi>"
 psk="<Password for your WiFi>"
}
```

Enabling SSH

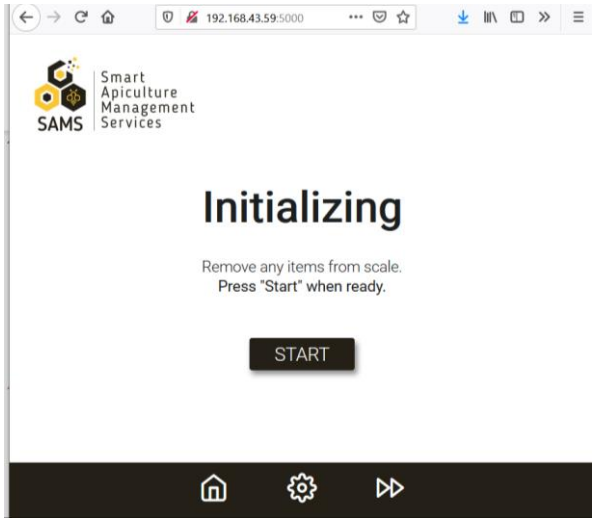
SSH can be enabled by placing a file called ssh into the boot folder. This flags the Pi to enable the SSH system on the next boot.

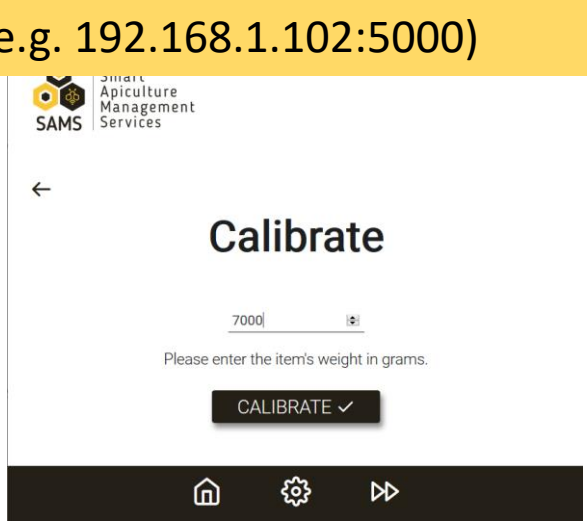
SAMS Monitoring System: Calibration

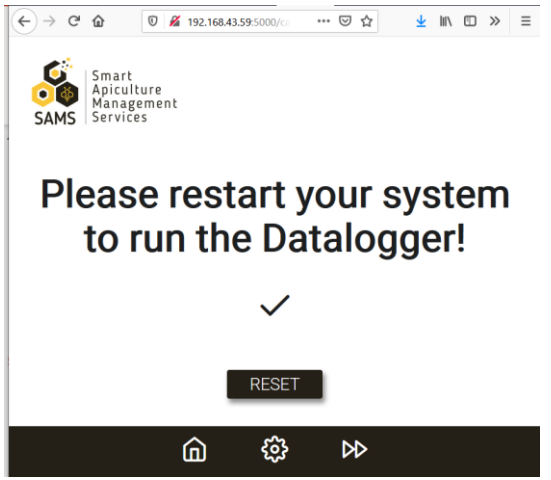
- Open Calibration Interface (Type Pi´s IP-Adress:5000 in Browser) (e.g. 192.168.1.102:5000)
- Follow Calibration Instructions

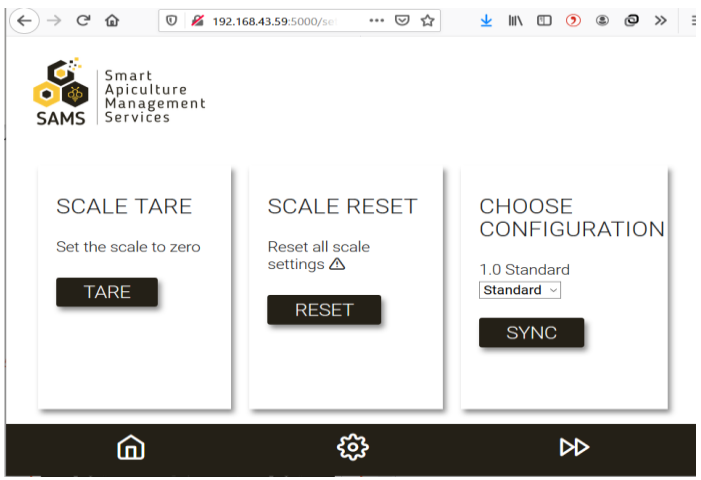
Calibrate your SAMS System

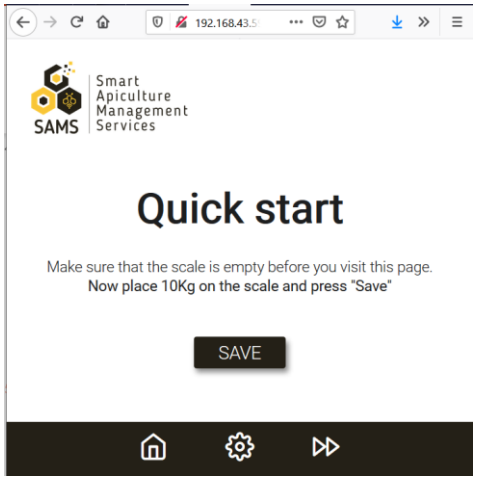
Open Calibration Interface in Browser
(Type „IP-Adress:5000“ - e.g. 192.168.1.102:5000)











SAMS System Test

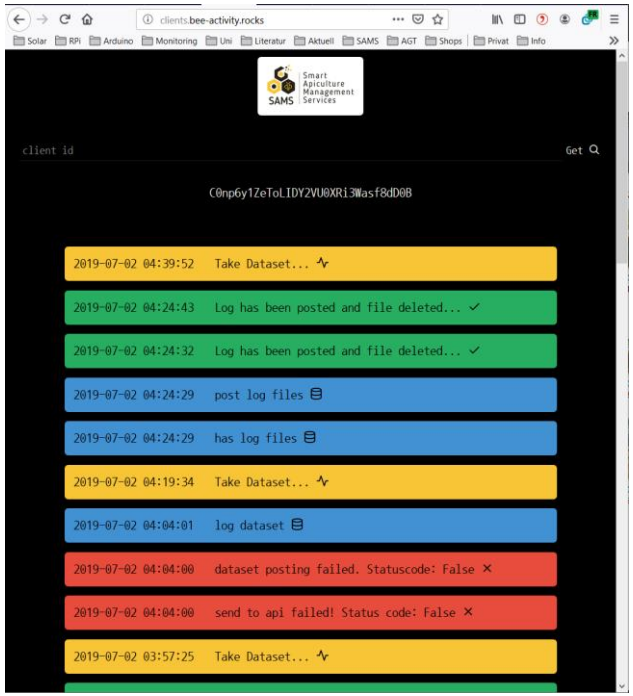
Check your System after Reboot:

- Via SSH:

  cd sams-config/data_logger

  python3 test.py

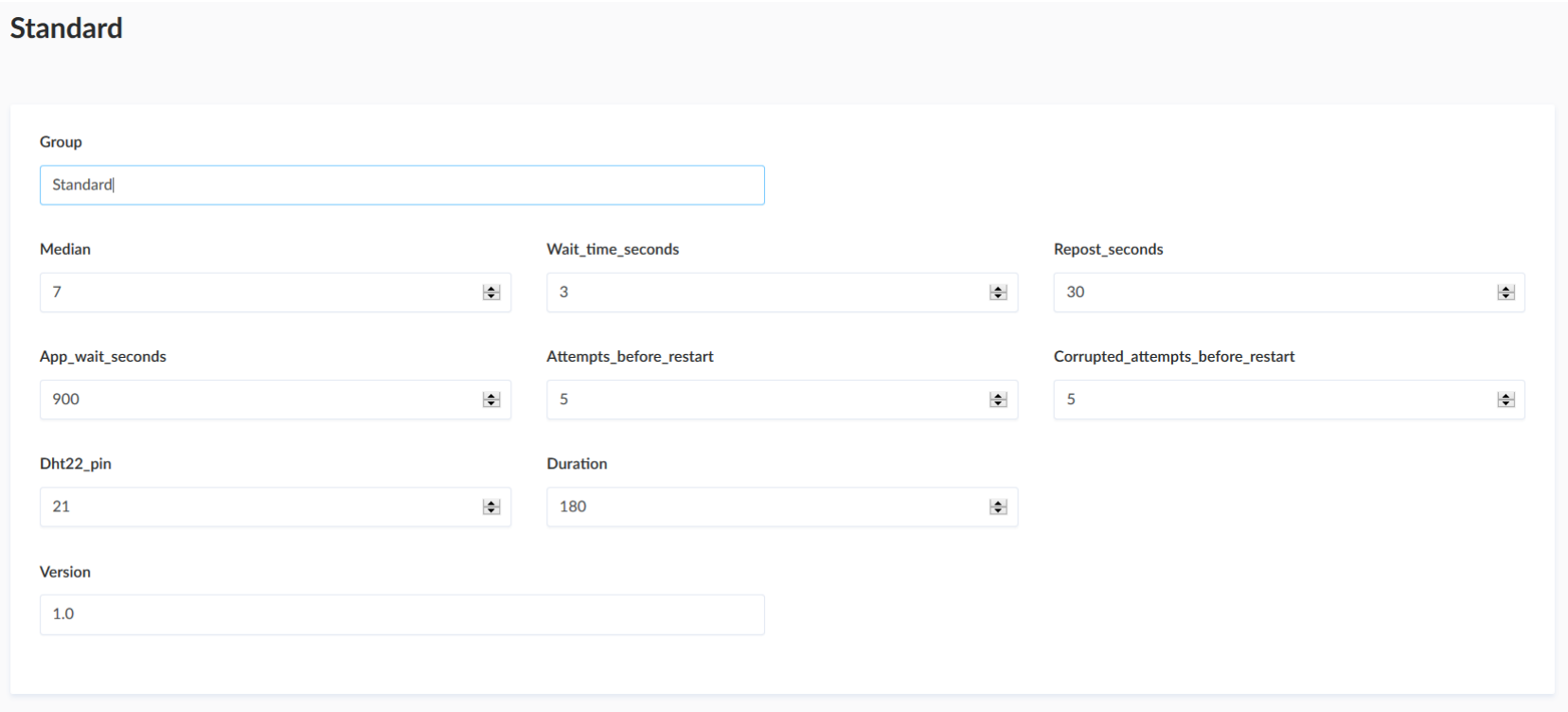
- Via Browser *IP-Adress:5000/api*
  (e.g. 192.168.0.102:5000/api)
- System LOG
  http://clients.bee-activity.rocks/
  Use your specific *client ID*



Smart Apiculture Management Services

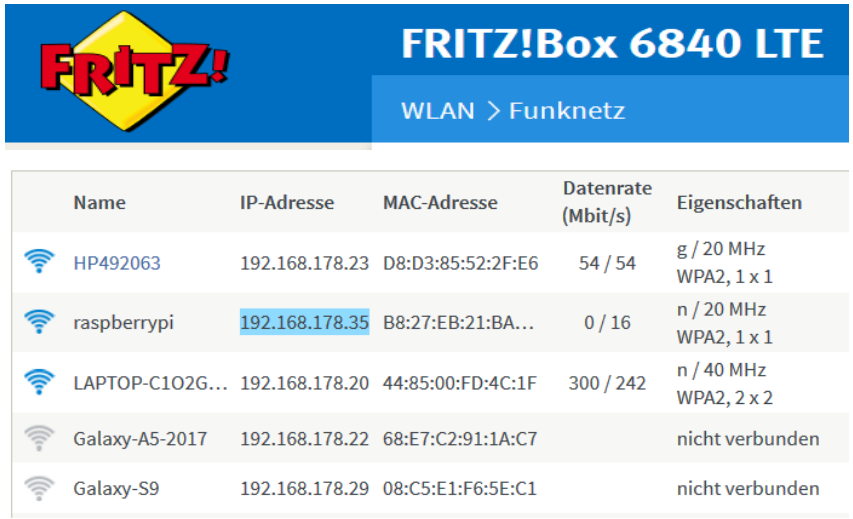
Online .config

/sams_system/data-logger/config_plugin (via SSH)



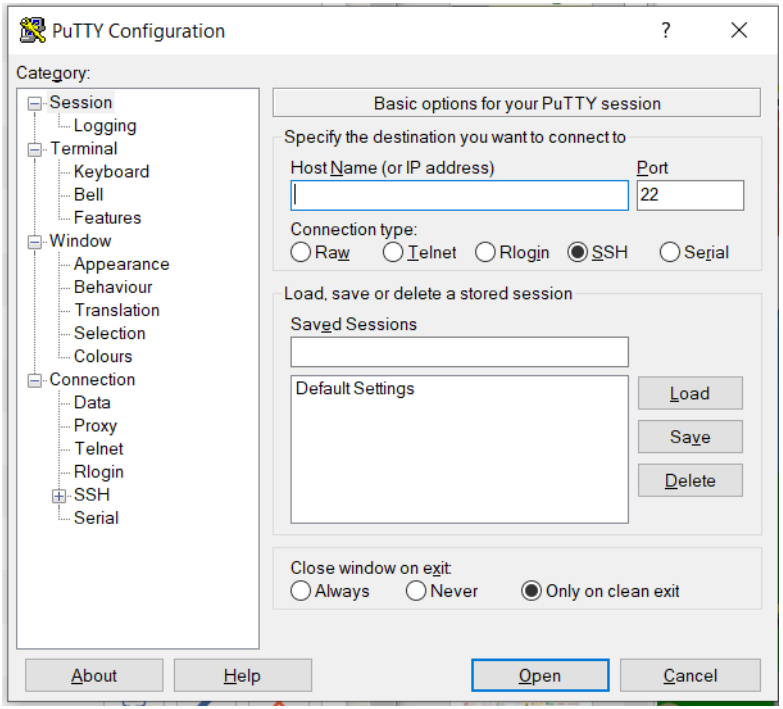
Connect via SSH to your Pi

Find your Pi´s IP-Adress on the Router



	Name	IP-Adresse	MAC-Adresse	Datenrate (Mbit/s)	Eigenschaften
	HP492063	192.168.178.23	D8:D3:85:52:2F:E6	54 / 54	g / 20 MHz WPA2, 1 x 1
	raspberrypi	192.168.178.35	B8:27:EB:21:BA…	0 / 16	n / 20 MHz WPA2, 1 x 1
	LAPTOP-C1O2G…	192.168.178.20	44:85:00:FD:4C:1F	300 / 242	n / 40 MHz WPA2, 2 x 2
	Galaxy-A5-2017	192.168.178.22	68:E7:C2:91:1A:C7		nicht verbunden
	Galaxy-S9	192.168.178.29	08:C5:E1:F6:5E:C1		nicht verbunden

SSH Connection with „PuTTY“



Login: pi
Standard Password: raspberry
SAMS Password: samsrocks

Data Warehouse: Credentials.ini File

- Auth0 Authentification
- 1 *Client ID* + *Secret* for each Raspberry Pi
- Sensor ID mapping via Data Warehouse and *Credentials.ini* File in boot Folder:

```
[DEFAULT]
client_id = Zy9Jtu1uXEIw94OP3…OAYXWT
client_secret = Y5G1LTWaZh-P77A…WPdW6GKnd0
audience = sams-dwh-web-api
grant_type = client_credentials
data_url = https://sams.science.itf.llu.lv/api/data
token_url= https://sams-project.eu.auth0.com/oauth/token
ondo_key= 666
```

Place your specific *Client ID* and *Secret* here!

Smart Apiculture Management Services

SAMS Monitoring System: Configuration Steps and Tasks

- Build the Fritzing Circuit for SAMS System on the Breadboards
- Download and install Win32 Disk Imager: https://sourceforge.net/projects/win32diskimager/
- Download and install PuTTY: https://www.putty.org/
- Download and flash Image (*sams.img*) to SD-Card (Win32 DI)
- Install BerryLan App on your Smartphone: http://www.berrylan.org/
- Get WiFi Connection to your Raspberry Pi (Try BerryLan App!)
- Prepare *Credentials.ini* and copy into boot folder (Use your own *Client ID* and *Secret*!)
- Configure Data Warehouse (Use manual!)
- Calibrate SAMS System via Browser and Pi´s IP-Adress:5000 (e.g. 192.168.8.102:5000)

Configuration of the Data Warehouse

See Manual: SAMS_Data Warehouse.pdf

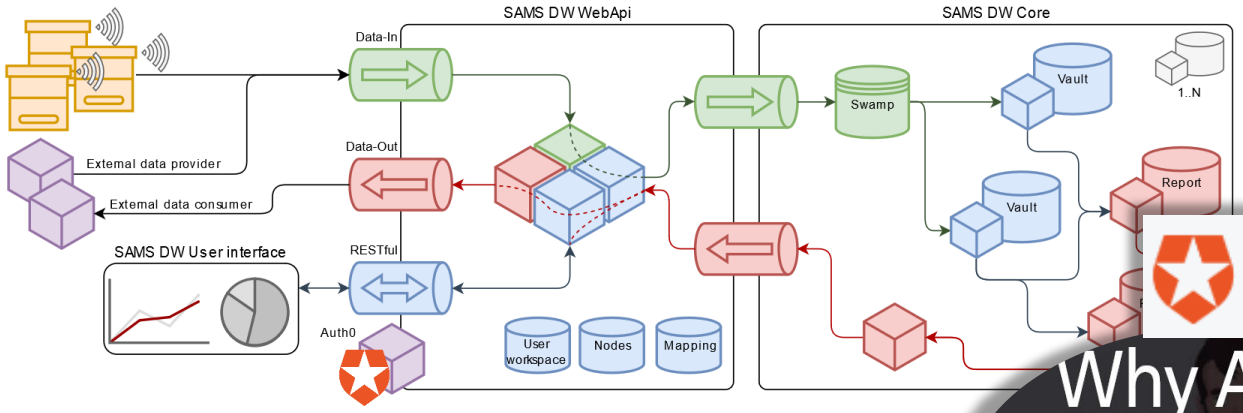
https://sams.science.itf.llu.lv

SAMS DW Dashboard Nodes Devices Reports Calculators didit@labtekindie.com

Welcome to Data Warehouse developed within SAMS project

SAMS is a project funded by the European Union within the H2020-ICT-39-2016-2017 call. SAMS enhances international cooperation of ICT (Information and Communication Technologies) and sustainable agriculture between EU and developing countries in pursuit of the EU commitment to the UN Sustainable Development Goal "End hunger, achieve food security and improved nutrition and promote sustainable agriculture".
To get more information about the SAMS project please visit: https://sams-project.eu/

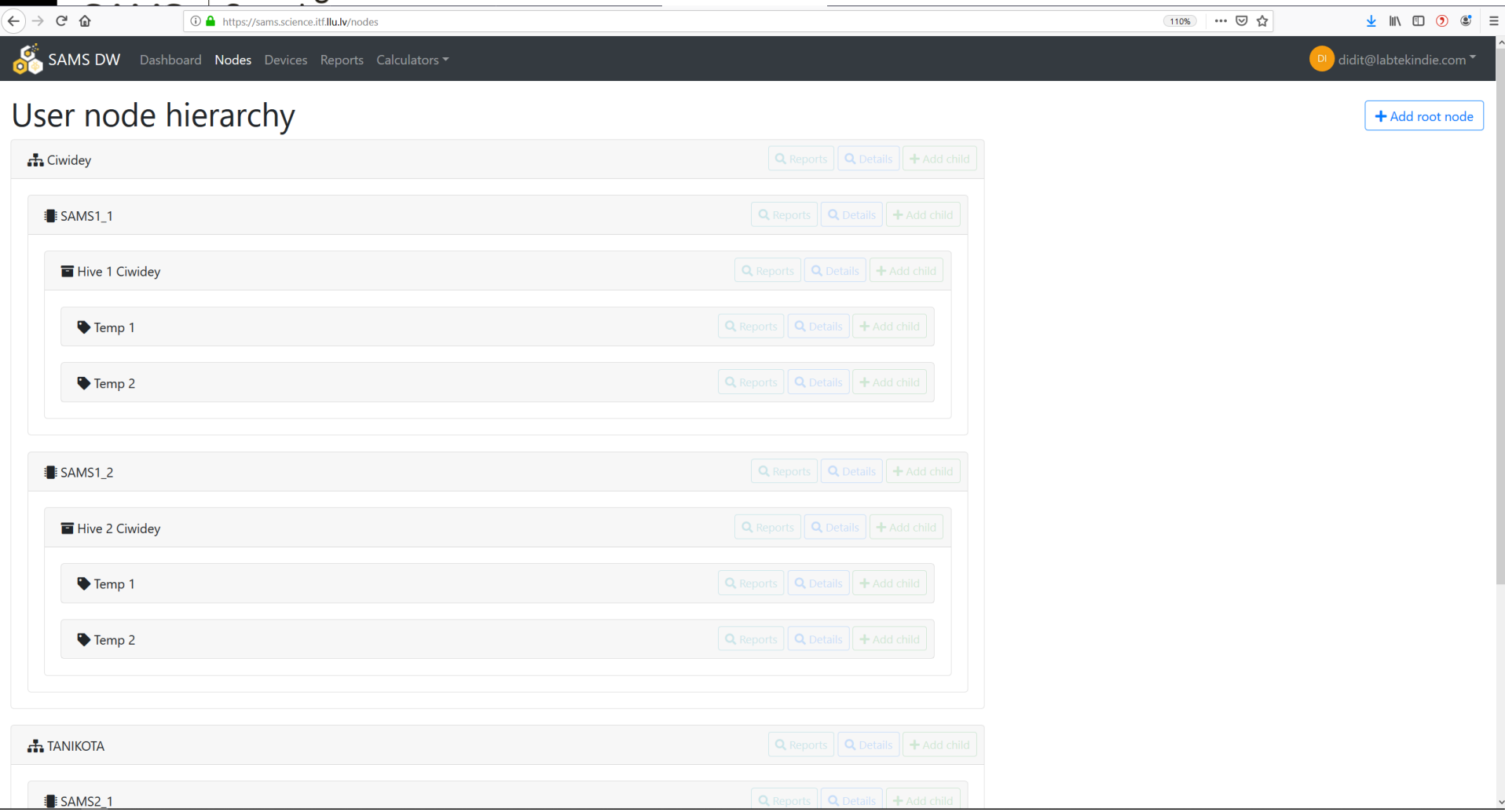
This Data Warehouse (DW) is a universal system, which is able to operate with different data inputs and have flexible data processing algorithms. Data warehouse archictecture is demonstrated below:

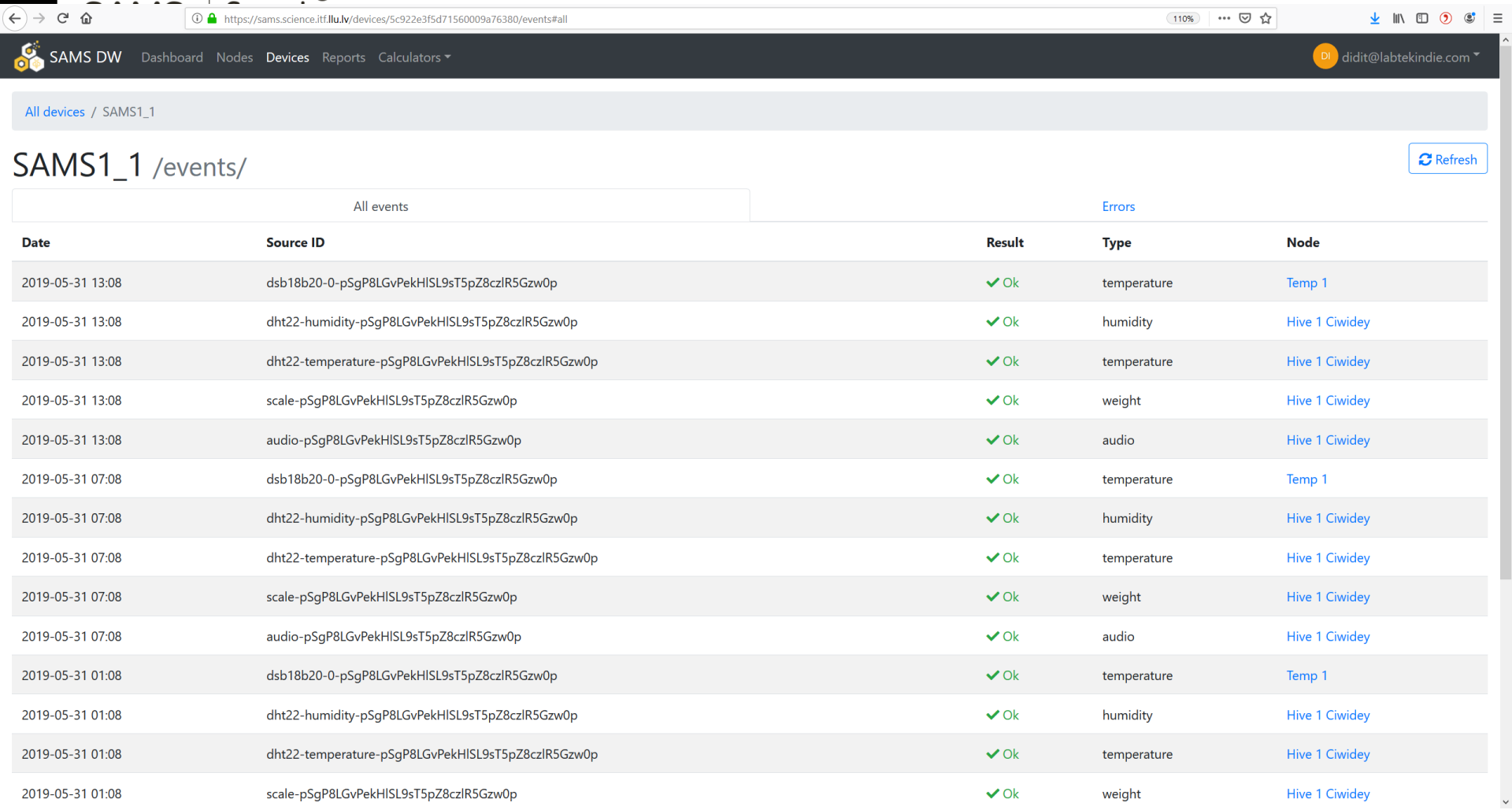


DW consists of three modules: Core, WebApi, GUI.
Benefit of the developed DW is that data are processed almost immediately by involving different models for data aggregation and repo
development considerations.

This project receives funding from the German Federal Ministry for Economic Cooperation and Development

This project receives funding from the Horizon 2020 European Union Research and Innovation Framework under Grant Agreement N° 780755 - SAMS


https://sams.science.itf.llu.lv/nodes
110%
SAMS DW
Dashboard
Nodes
Devices
Reports
Calculators
didit@labtekindie.com
User node hierarchy
+ Add root node
Ciwidey
Reports
Details
+ Add child
SAMS1_1
Hive 1 Ciwidey
Temp 1
Temp 2
SAMS1_2
Hive 2 Ciwidey
Temp 1
Temp 2
TANIKOTA
SAMS2_1



https://sams.science.itf.llu.lv/devices/5c922e3f5d71560009a76380/events#all

SAMS DW Dashboard Nodes Devices Reports Calculators didit@labtekindie.com

All devices / SAMS1_1

SAMS1_1 /events/

Refresh

All events Errors

Date	Source ID	Result	Type	Node
2019-05-31 13:08	dsb18b20-0-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	temperature	Temp 1
2019-05-31 13:08	dht22-humidity-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	humidity	Hive 1 Ciwidey
2019-05-31 13:08	dht22-temperature-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	temperature	Hive 1 Ciwidey
2019-05-31 13:08	scale-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	weight	Hive 1 Ciwidey
2019-05-31 13:08	audio-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	audio	Hive 1 Ciwidey
2019-05-31 07:08	dsb18b20-0-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	temperature	Temp 1
2019-05-31 07:08	dht22-humidity-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	humidity	Hive 1 Ciwidey
2019-05-31 07:08	dht22-temperature-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	temperature	Hive 1 Ciwidey
2019-05-31 07:08	scale-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	weight	Hive 1 Ciwidey
2019-05-31 07:08	audio-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	audio	Hive 1 Ciwidey
2019-05-31 01:08	dsb18b20-0-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	temperature	Temp 1
2019-05-31 01:08	dht22-humidity-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	humidity	Hive 1 Ciwidey
2019-05-31 01:08	dht22-temperature-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	temperature	Hive 1 Ciwidey
2019-05-31 01:08	scale-pSgP8LGvPekHlSL9sT5pZ8czlR5Gzw0p	✔ Ok	weight	Hive 1 Ciwidey

SAMS Smart Apiculture Management Services

RTC + Power Management with Witty Pi Mini

http://www.uugear.com/product/wittypi-mini/ UUGear

Witty Pi Mini adds RTC (DS3231SN) and power management to your Raspberry Pi, and can define complex ON/OFF sequence with simple script.

After installing the software on your Raspberry Pi, you can use these features:

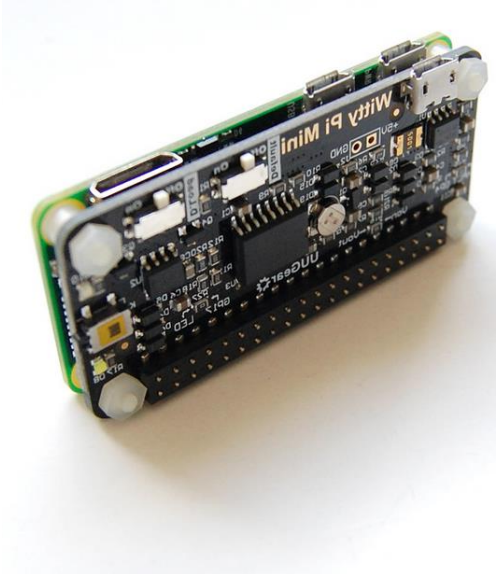
- Single tap on the button to turn on/off Raspberry Pi.
- After shutdown, the power for Raspberry Pi and all its USB peripherals are fully cut.
- Raspberry Pi knows the correct time, even without accessing the Internet.
- You can schedule the startup/shutdown of your Raspberry Pi.
- You can even write a script to define complex ON/OFF sequence.

See user manual: http://www.uugear.com/doc/WittyPiMini_UserManual.pdf

Script Generator: http://www.uugear.com/app/wittypi-scriptgen/

SAMS Smart Apiculture Management Services

Witty Pi Mini Setup



To install the software, just run these two commands:

wget http://www.uugear.com/repo/WittyPi2/installWittyPi.sh

sudo sh installWittyPi.sh

Remove fake-hwclock package and disable ntpd daemon? (recommended) [y/n] y

Do you want to install Qt 5 for GUI running? [y/n] n

After the installation is done, turn off your Raspberry Pi:

sudo poweroff

Then physically mount Witty Pi Mini on it. Connect 5V DC power supply to the micro USB port on Witty Pi Mini, and tap the button to start. After login to the system, just run these commands to launch the Witty Pi utility:

cd wittyPi

sudo ./wittyPi.sh

After Installation and Connection of WittyPi: Access and configure it!

Access wittyPi

1. *cd wittyPi*
2. *sudo ./wittyPi.sh*
3. Set system time to RTC time (*1*)
4. set scheduleing script (*6*)
5. Select scheduling script (*1-x*)
6. Leave wittyPi-Menu (*8*)
7. Dissconect from Device: *exit*

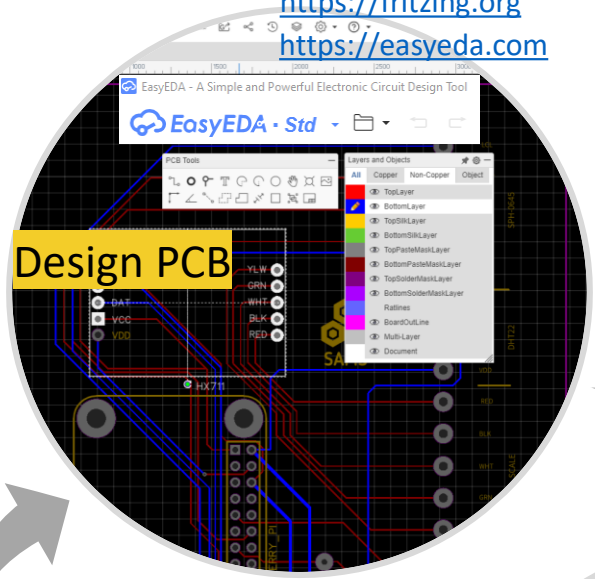
Adapt scheduling scripts

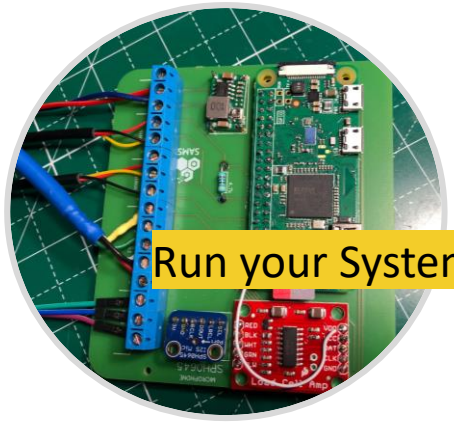
The scheduling scripts are stored in the directory *wittyPi/schedules*. Thus,

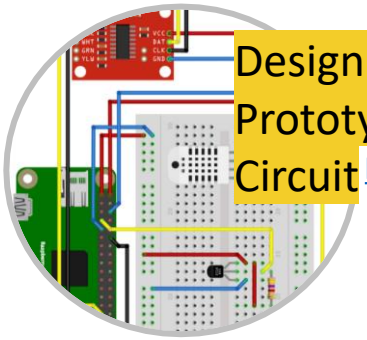
1. *cd wittyPi*
2. *cd schedules*
3. Alternatively *cd wittyPi/schedules*,
4. to return to a parent directory (moving upwards), type *cd ..*
5. to list the available scripts, type *ls* or *dir*,
6. to edit scheduling script *sams.wpi*, type *sudo nano sams.wpi*
7. the nano-editor opens, the *sams.wpi* scipt is empty

Design your Printed Circuit Board (PCB)

https://fritzing.org
https://easyeda.com

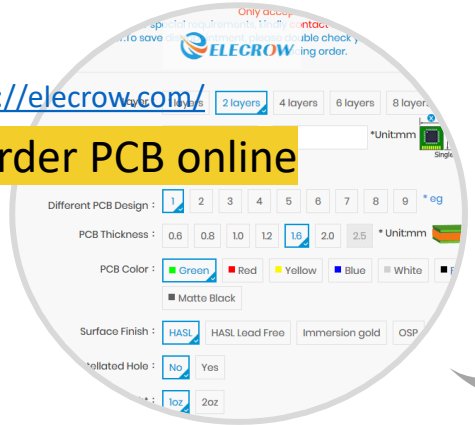


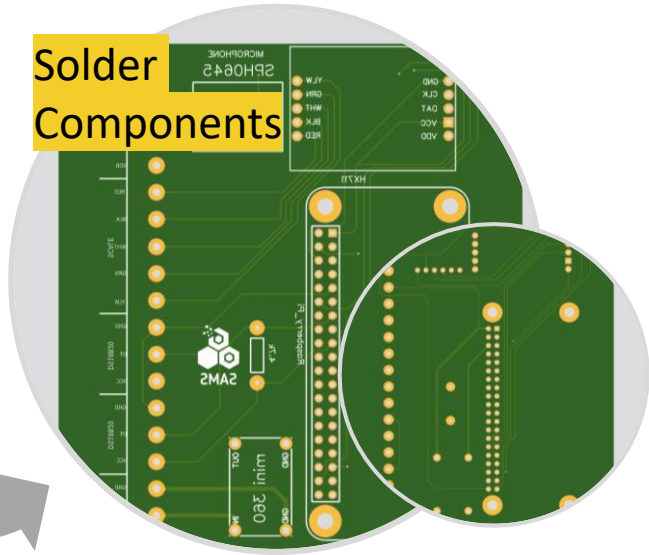




https://fritzing.org

https://elecrow.com/

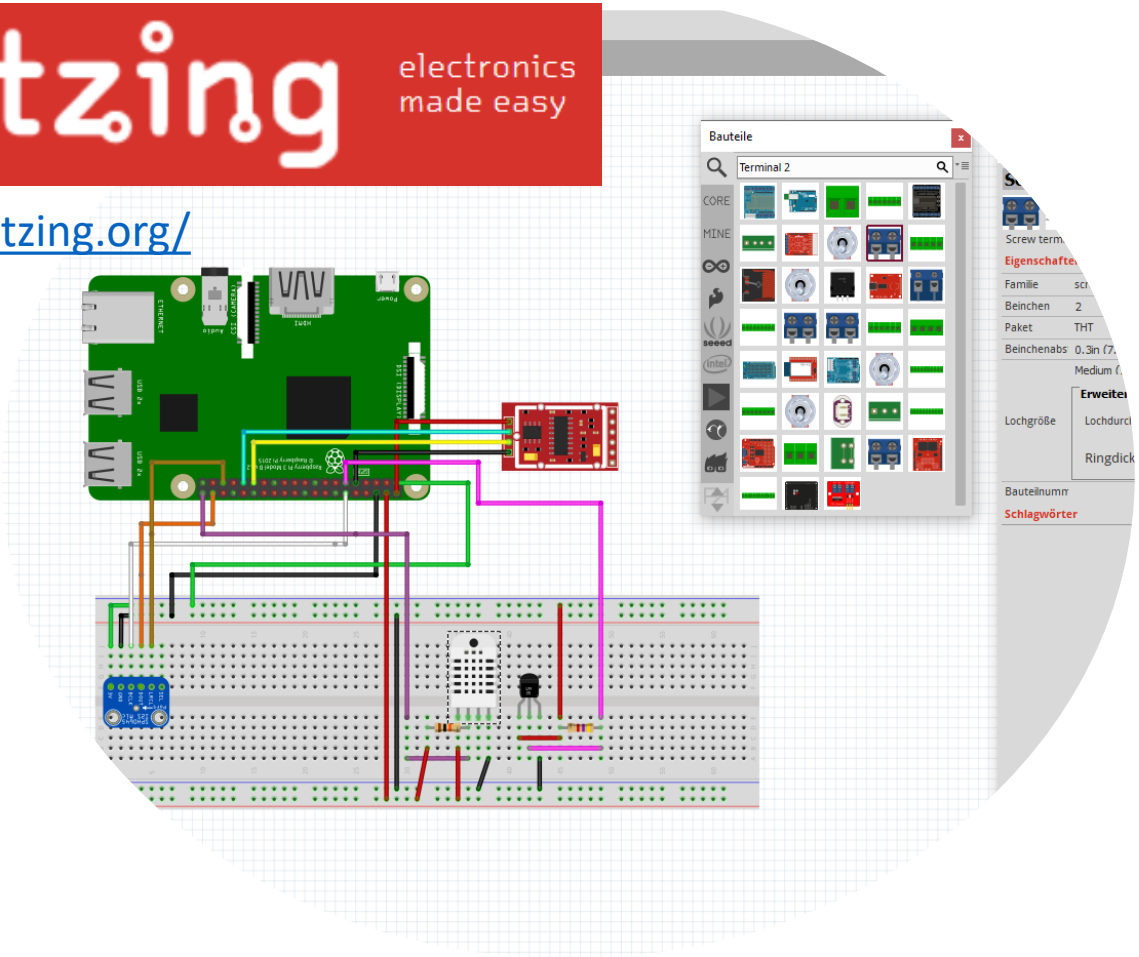




Design with Fritzing

fritzing electronics made easy

https://fritzing.org/



Other useful tasks to try out

- Try to use Rasbian desktop full version on your Pi with Screen (https://www.raspberrypi.org)
- Try different ways to get WiFi Connection to your Pi
- Try to connect your Pi with your Computer via SSH (e.g. puTTY)
- Try to work with Fritzing (Build your own SAMS Circuit Plan and PCB)
  - Try to place Components at its best
- Try to install and configure WittyPi Mini and use your own schedule Script (you can also use standard Raspbian Image)
- Try to make your own python Script to get data from temperature sensor (Use manual on Slide 34)

1-Wire Interface on the Pi

The Raspberry Pi has a 1-wire bus which can be enabled on GPIO4. It provides low-speed data, signaling, and power over a single conductor. This is commonly used on the Pi to connect low-cost sensors devices such as the DS18B20 temperature sensor. Several sensors can be added.

To enable the interface on the Pi we need to make a modification to a system file (or using 'Interfacing Options' within the raspi-config utility: *sudo raspi-config*):

sudo nano /boot/config.txt

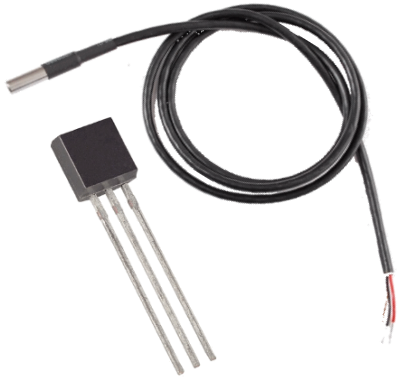
Add the following line at the bottom :

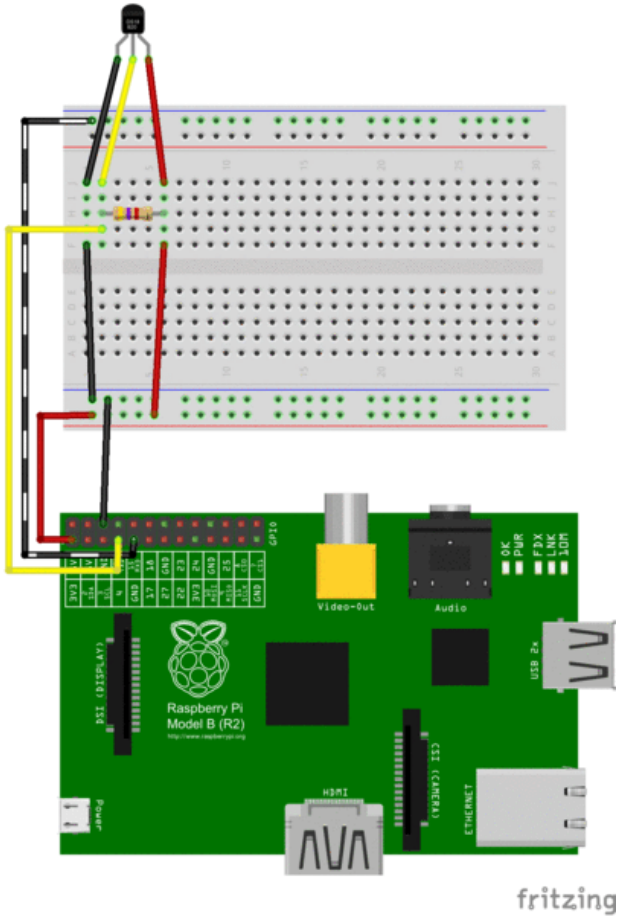
dtoverlay=w1-gpio

Use CTRL-X, then Y, then RETURN to save the file and exit.

This will use Pin 7 (GPIO4) by default.

Example: Temperature with the 1-Wire interface and the DS18B20





Building the Circuit
You will need:

- A Dallas DS18B20 (either version)
- 4.7k ohm Resistor
- 4 female to male jumper wires
- 2 male to male jumper wires

[See Online Tutorial:
http://www.circuitbasics.com/](http://www.circuitbasics.com/)

Temperature with the 1-Wire interface and the DS18B20: Configuration

Configuring your Raspberry Pi

Before you can use any 1-wire devices you must first tell the Raspberry Pi how to read them. Open a Terminal Window and type the following to edit the Raspberry Pi's configuration file:

sudo nano /boot/config.txt

Look to see whether there is a line that has *dtoverlay=w1-gpio* in it. If not, add the following to the end of the file:

dtoverlay=w1-gpio

Now reboot the Raspberry Pi:

sudo reboot

Temperature with the 1-Wire interface and the DS18B20: Configuration

To test the configuration, set-up the circuit above to connect the DS18B20 and type the following into a terminal window:

sudo modprobe w1-gpio
sudo modprobe w1-therm
cd /sys/bus/w1/devices
ls

This will list all the devices that are connected to the 1-wire interface. The Dallas DS18B20 sensor starts with '28-' followed by a long number. Type in the following, replacing the 'xxxx' with the text following the '28-':

cd 28-xxxx
cat w1_slave

In response, you should get the following showing that the DS18B20 is working:

a3 01 4b 46 7f ff 0e 10 d8 : crc=d8 YES
a3 01 4b 46 7f ff 0e 10 d8 t=32768

Temperature with the 1-Wire interface and the DS18B20: Python Script

Installs or updates Python 3:

```
sudo apt-get install python3
```

Reading from the Sensor

Create a new python script from a terminal window (with sudo nano temperature.py). Enter the following example code:

```
# Import Libraries
import os
import glob
import time

# Initialize the GPIO Pins
os.system('modprobe w1-gpio')  # Turns on the GPIO module
os.system('modprobe w1-therm') # Turns on the Temperature module

# Finds the correct device file that holds the temperature data
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

# A function that reads the sensors data
def read_temp_raw():
  f = open(device_file, 'r') # Opens the temperature device file
  lines = f.readlines() # Returns the text
  f.close()
  return lines

# Convert the value of the sensor into a temperature
def read_temp():
  lines = read_temp_raw() # Read the temperature 'device file'
  # While the first line does not contain 'YES', wait for 0.2s
  # and then read the device file again.
  while lines[0].strip()[-3:] != 'YES':
    time.sleep(0.2)
    lines = read_temp_raw()

  # Look for the position of the '=' in the second line of the
  # device file.
  equals_pos = lines[1].find('t=')

  # If the '=' is found, convert the rest of the line after the
  # '=' into degrees Celsius, then degrees Fahrenheit
  if equals_pos != -1:
    temp_string = lines[1][equals_pos+2:]
    temp_c = float(temp_string) / 1000.0
    temp_f = temp_c * 9.0 / 5.0 + 32.0
    return temp_c, temp_f

# Print out the temperature until the program is stopped.
while True:
  print(read_temp())
  time.sleep(1)
```

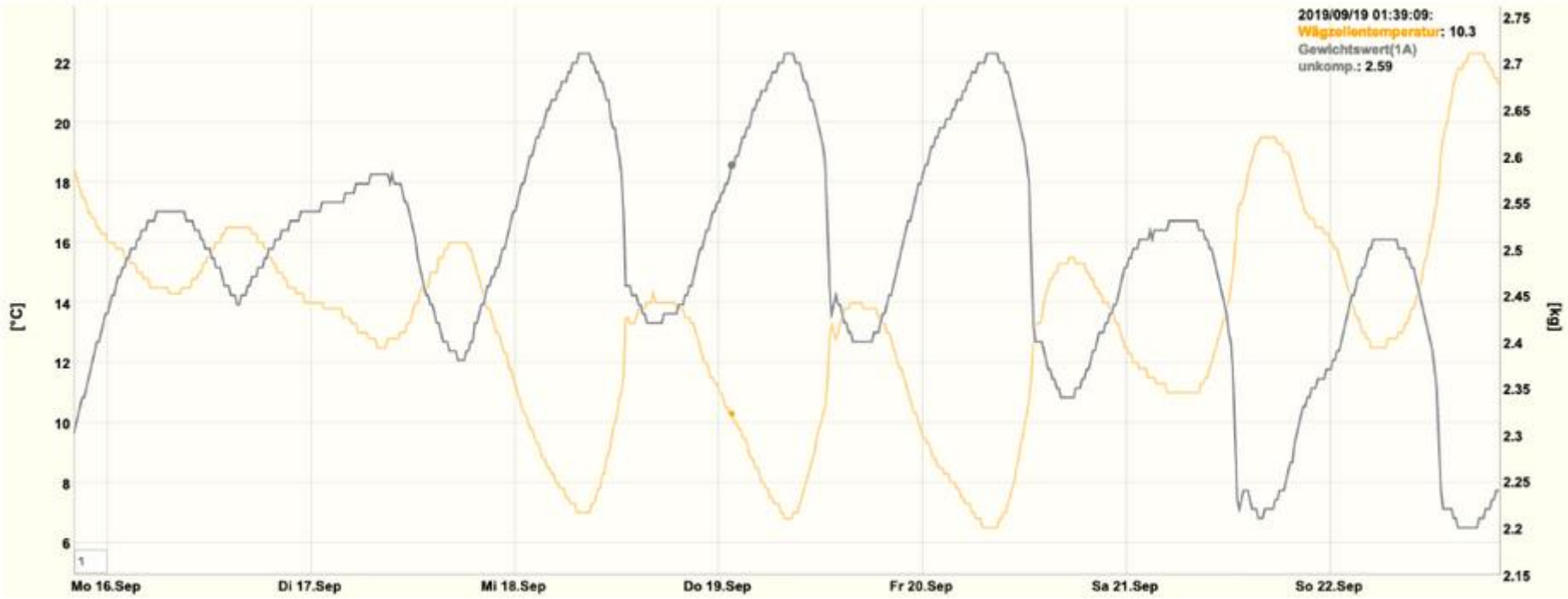
To run the code, you need to be the Super User, so run the code with:

```
sudo python temperature.py
```

SAMS Smart Apiculture Management Services

Theoretical Excursus: Temperature Compensation of Sensors (e.g. Load cell)

Example: Weight curve of a load cell (without temp. compensation) for a fixed weight of approx. 2.6kg



Smart Apiculture Management Services

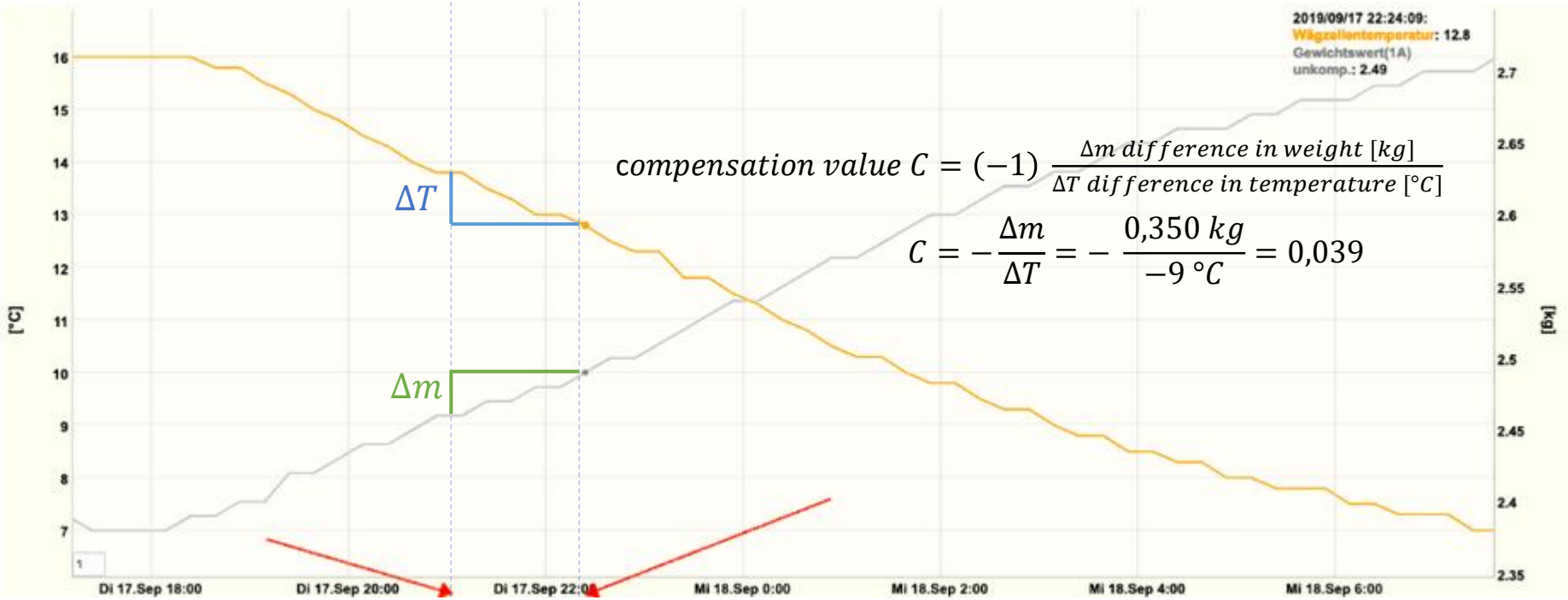
SAMS

Scale: Temperature Compensation

Choose one of the highest possible difference in the temperature curve

Scale: Temperature Compensation

Select a range between these values and calculate the temperature and weight difference, as well as the compensation value.



$$compensation\ value\ C = (-1)\ \frac{\Delta m\ difference\ in\ weight\ [kg]}{\Delta T\ difference\ in\ temperature\ [°C]}$$

$$C = -\frac{\Delta m}{\Delta T} = -\frac{0{,}350\ kg}{-9\ °C} = 0{,}039$$

Smart Apiculture Management Services

SAMS

Challenges and limits

Technology is the answer, but what was the question?

- Cedric Price -

SAMS Smart Apiculture Management Services

Some Challenges and limits of digitalisation …

- **Social-ecological objectives**

  Digitalisation should contribute to sustainable energy, transport, agriculture and resource management (human rights, climate protection goals, ending hunger and poverty)

- **Privacy and control of monopolies**

  Framework conditions for the control of digital monopolies

- **Democracy**

  Democratic decisions are the basis of a fair society

- **Education**

  Critical and emancipatory handling of digital technology and media

- **Development and trade policy aspects**

  Trade agreements must not contain any restrictions for Taxation, Open Source and Localization. All countries should share the benefits and costs of digitisation equally.

- **IT security**

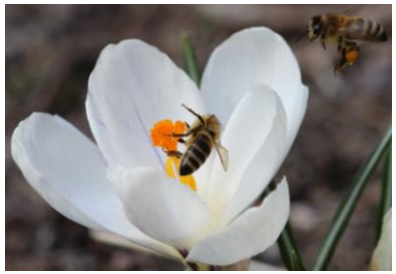
  Software liability: software developers need to take responsibility for the emerging risks

- **Durability of software and hardware**

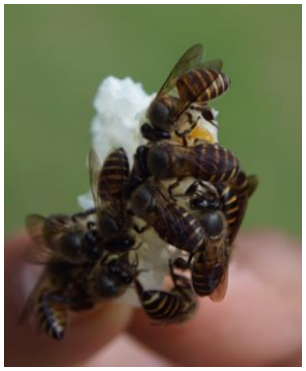
  Electronic devices and software must be repairable and recyclable - there must be no planned obsolescence

Source: https://bits-und-baeume.org/downloads/BUB_Forderungen_A5_Einzelseiten.pdf

This project has received funding from the European Union´s Horizon 2020 research and innovation programme under Grant Agreement N° 780755. The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the EU



Thank you!



For more information,
visit our website
www.sams-project.eu
or

contact the project coordinator

angela.zur@giz.de

or

follow us on Twitter:
@SAMS_EU_H2020