# ECE 441
# Smart & Connected Embedded Systems

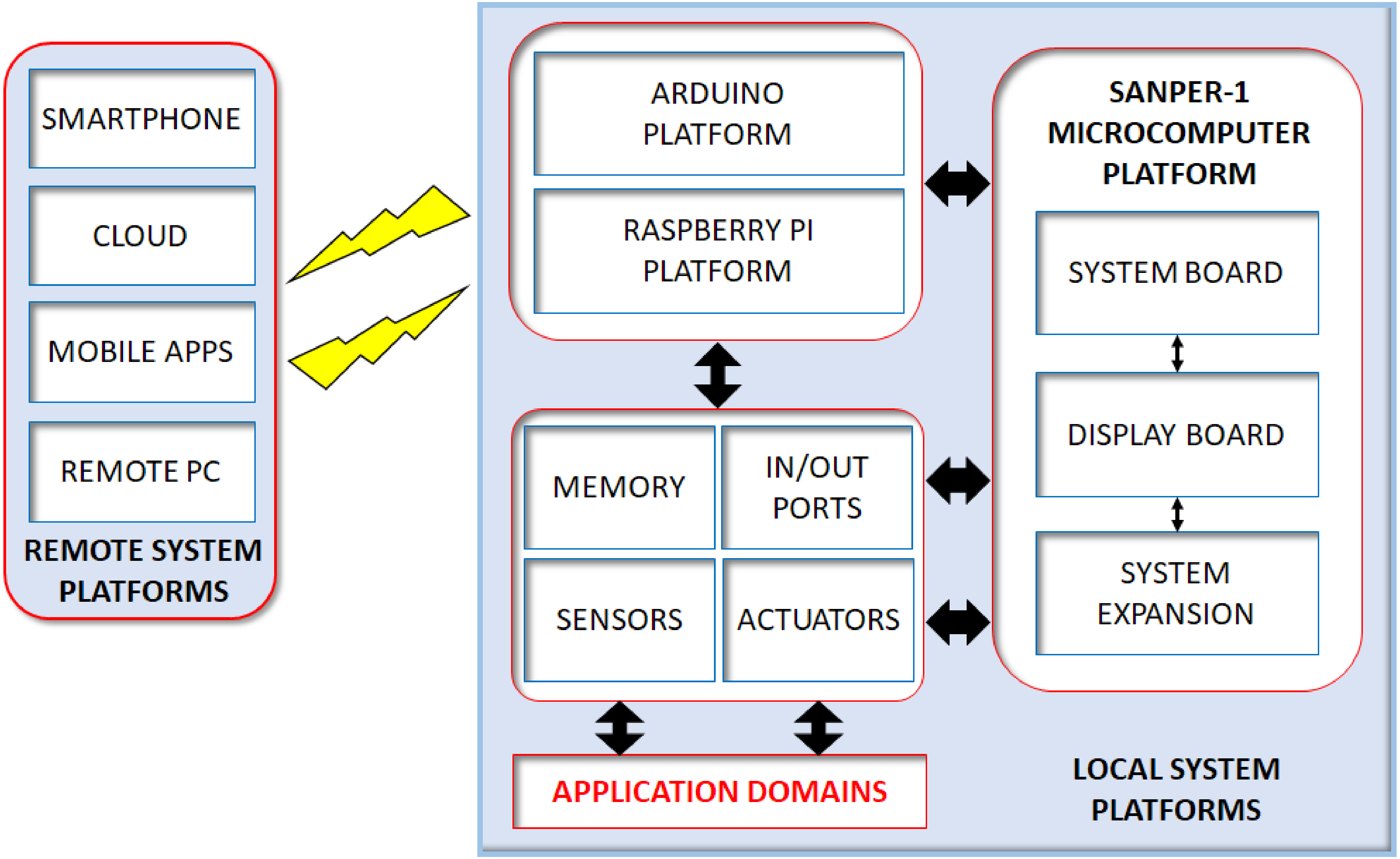
**Instructor:** Professor Jafar Saniie

**Teaching Assistants:** Xinrui Yu, Xin Huang, and Mikhail Gormov

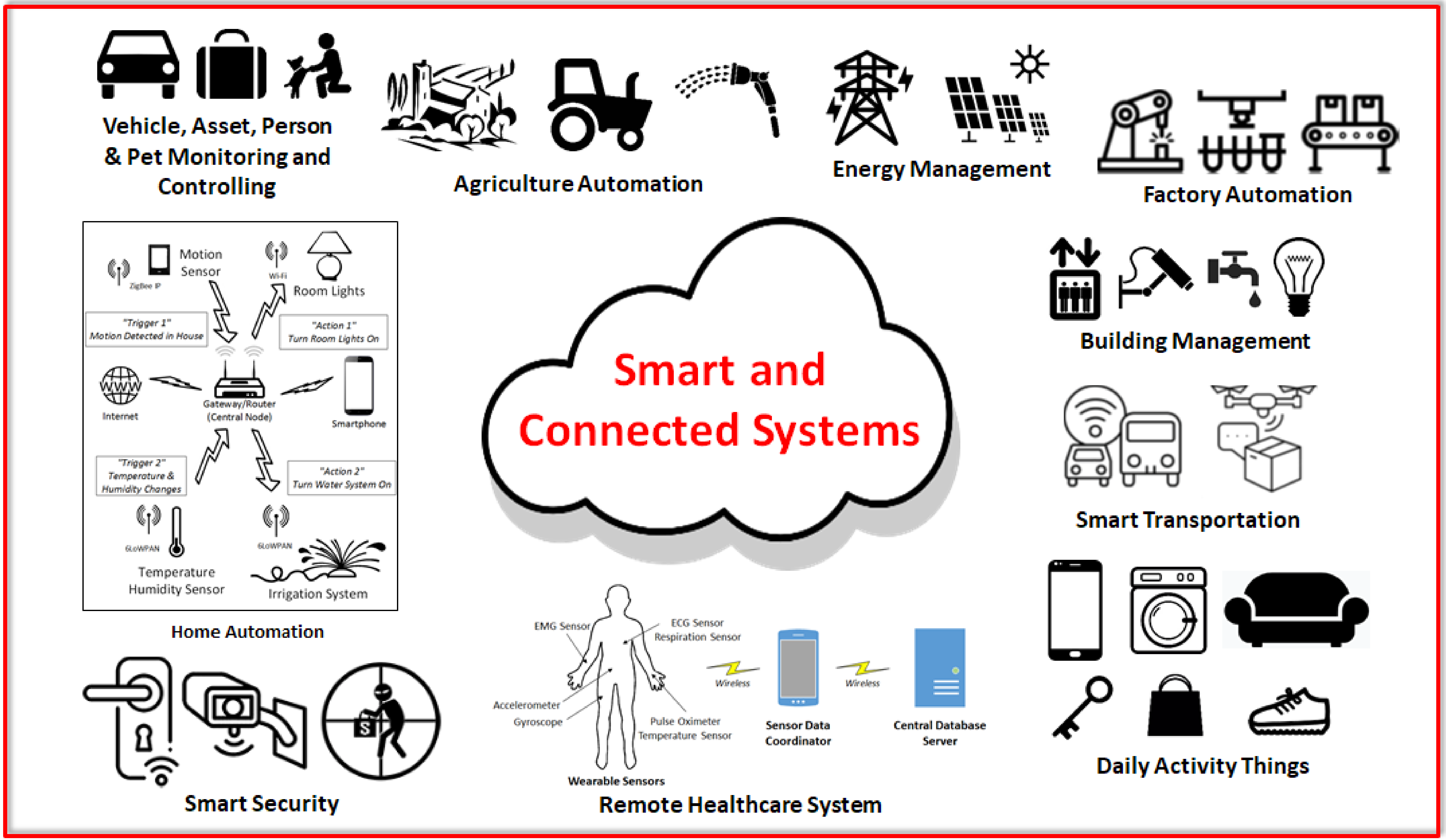
# Wide Range of Projects

- Multiple computing platform architectures for smart and connected system applications can be used.
- The goal is to develop smart and connected systems.
- Examples for the design project is shown later in this presentation.

# ECE 441 – MAJOR DESIGN PROJECT:
# SMART AND CONNECTED SYSTEM ARCHITECTURE

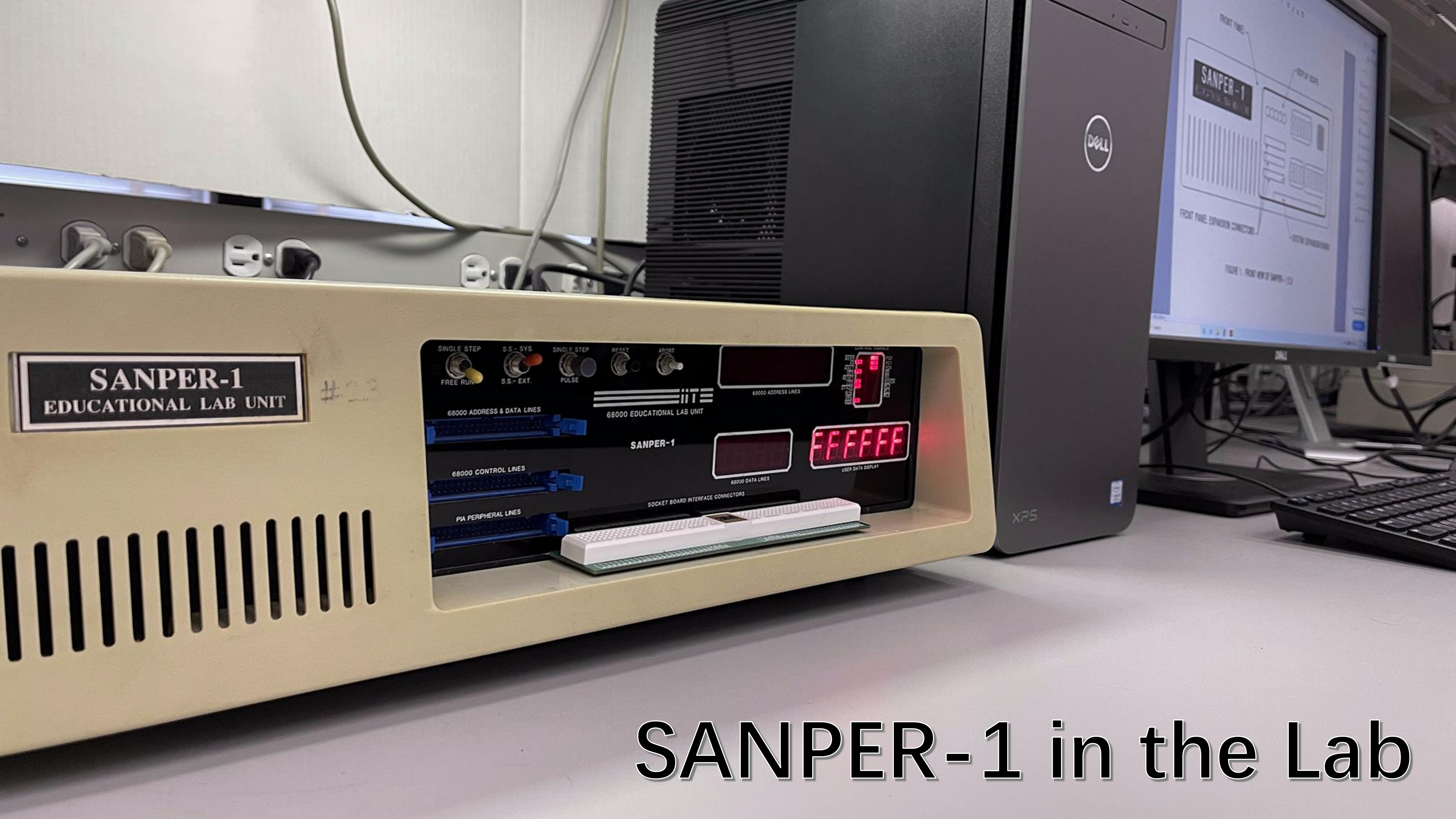


# APPLICATION DOMAINS



# What is SANPER?

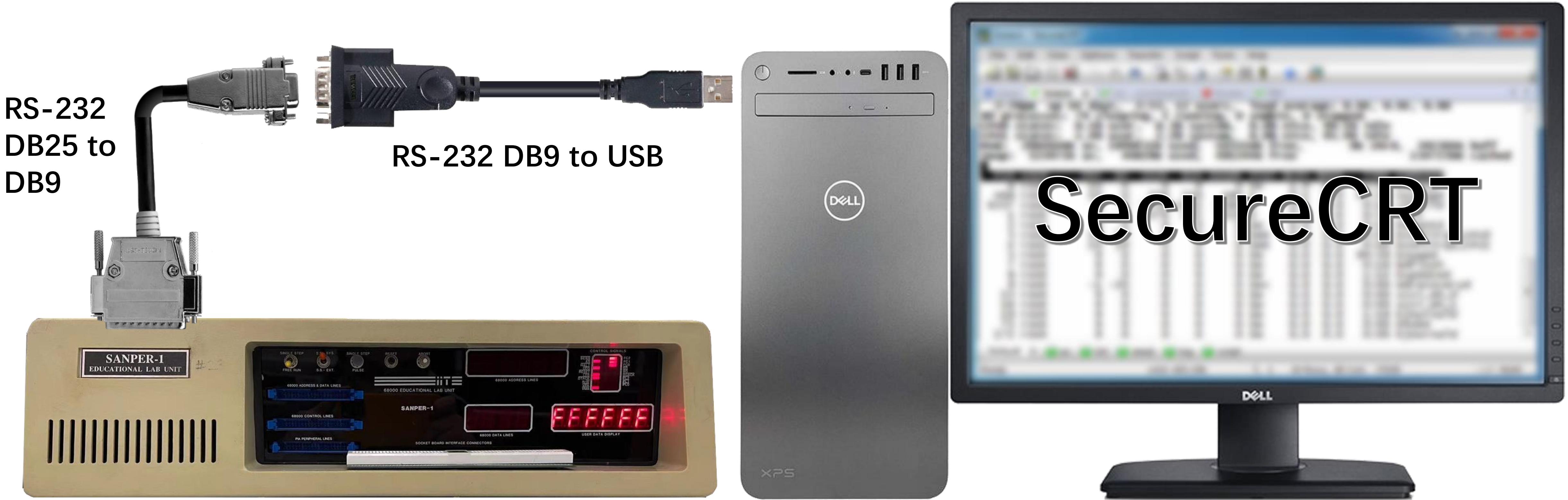
- The SANPER-1 Microcomputer Platform was designed and developed by Dr. Jafar Saniie and Mr. Stephen Perich.
- The platform includes a firmware package named “TUTOR” which is a MC68000 resident monitor program developed by Motorola, Inc.
- The platform could be used along with an external power supply and a video display terminal to construct an educational development system for the design project.



# SANPER-1 in the Lab

# Basic Access to SANPER

- The SANPER is connected to the lab PC via a RS-232 DB25 to DB9 cable and a RS-232 DB9 to USB cable.
- The lab PC runs SecureCRT (a terminal emulator) to communicate with SANPER.

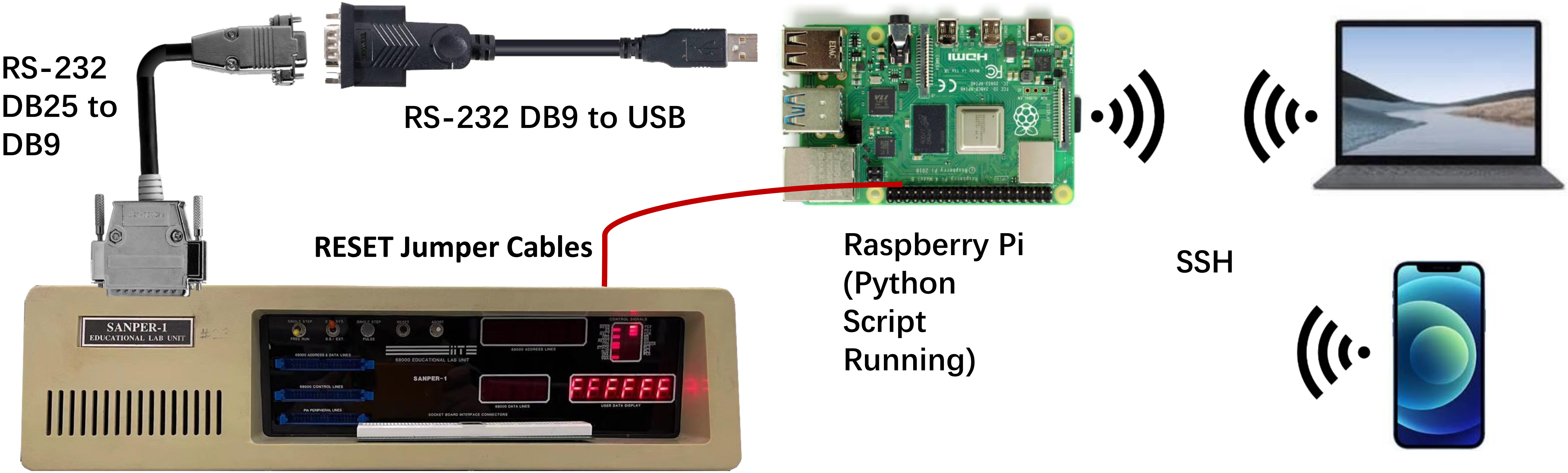


# Remote Access to SANPER

- Uses Raspberry Pi as a terminal to access SANPER remotely
- Access SANPER in a way like using SecureCRT on lab PCs
- Hardware RESET available without physically pressing the button
- Access SANPER remotely using PCs and smartphones
- **Hardware:**
  - **Raspberry Pi**
  - **RS-232 DB25 to DB9 cable**
  - **RS-232 DB9 to USB cable**
  - **Jumper cables**

# Access SANPER Remotely using Raspberry Pi

- The SANPER is connected to the Raspberry Pi with the same configuration.
- The Raspberry Pi runs a custom-built Python Script to communicate with SANPER.
- The Raspberry Pi can then communicate with other PCs and smartphones over the internet with SSH (Secure Shell, a cryptographic network protocol).



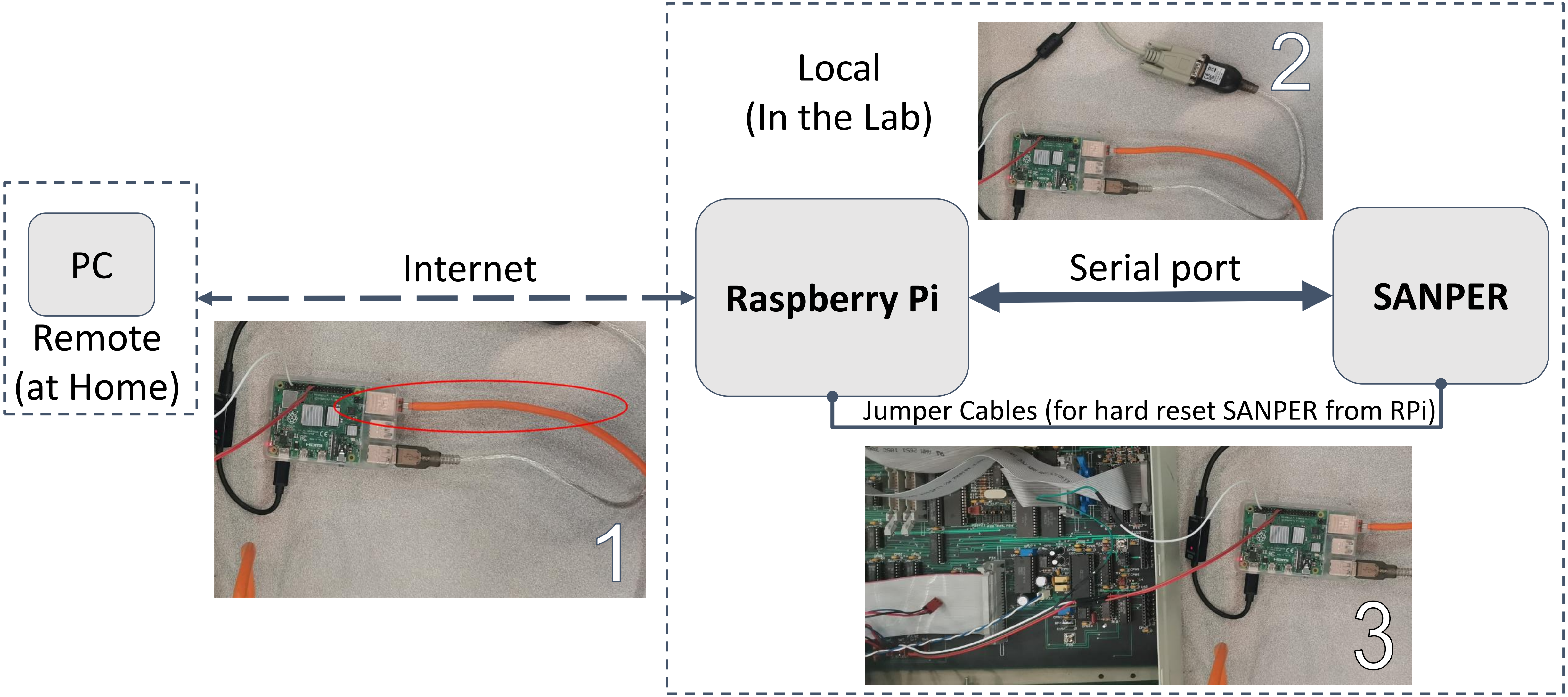
RS-232 DB25 to DB9

RS-232 DB9 to USB

RESET Jumper Cables

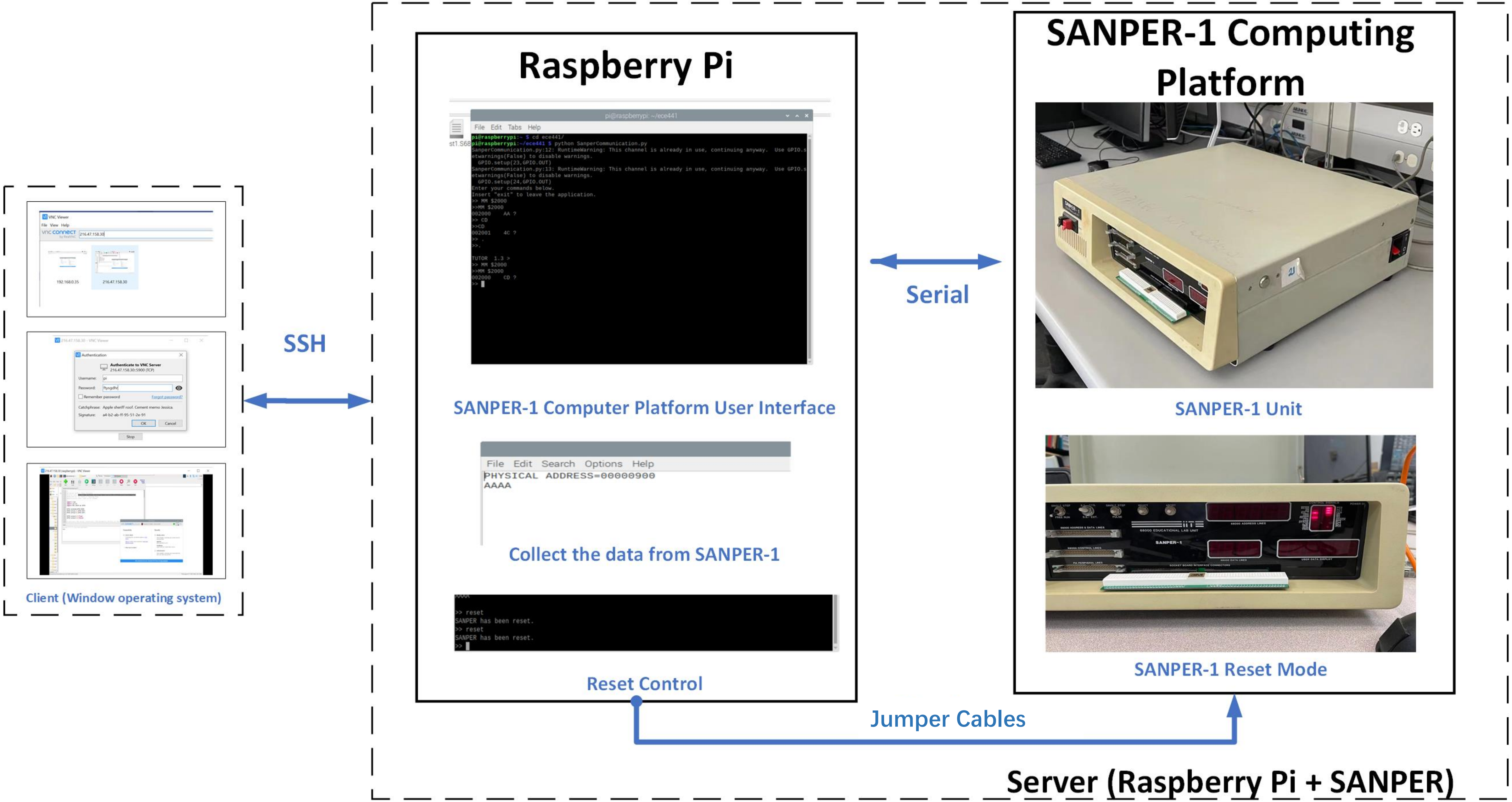
Raspberry Pi (Python Script Running)

SSH



- The Raspberry Pi connects to the Internet through the orange ethernet cable
- The silver cable is the serial port connection between SANPER-1 and Raspberry Pi
- Three jumper cables connect the RESET input of the SANPER-1 and three GPIO ports on the Raspberry Pi for remote hard reset of the SANPER-1.

# Visualization Interface of Remote Access to SANPER



Client (Window operating system)
SSH
Server (Raspberry Pi + SANPER)
Raspberry Pi
SANPER-1 Computer Platform User Interface
Collect the data from SANPER-1
Reset Control
Serial
SANPER-1 Computing Platform
SANPER-1 Unit
SANPER-1 Reset Mode
Jumper Cables
VNC Viewer
File View Help
vnc connect by RealVNC
192.168.0.35
216.47.158.30
216.47.158.30 - VNC Viewer
Authentication
OK
Cancel
Stop
216.47.158.30 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/ece441
File Edit Tabs Help
st1.S68
File Edit Search Options Help

VNC connect: 216.47.158.30

Authenticate to VNC Server
216.47.158.30::5900 (TCP)

Username: pi

Password: ftysgdhr

□ Remember password    Forgot password?

Catchphrase: Apple sheriff roof. Cement memo Jessica.

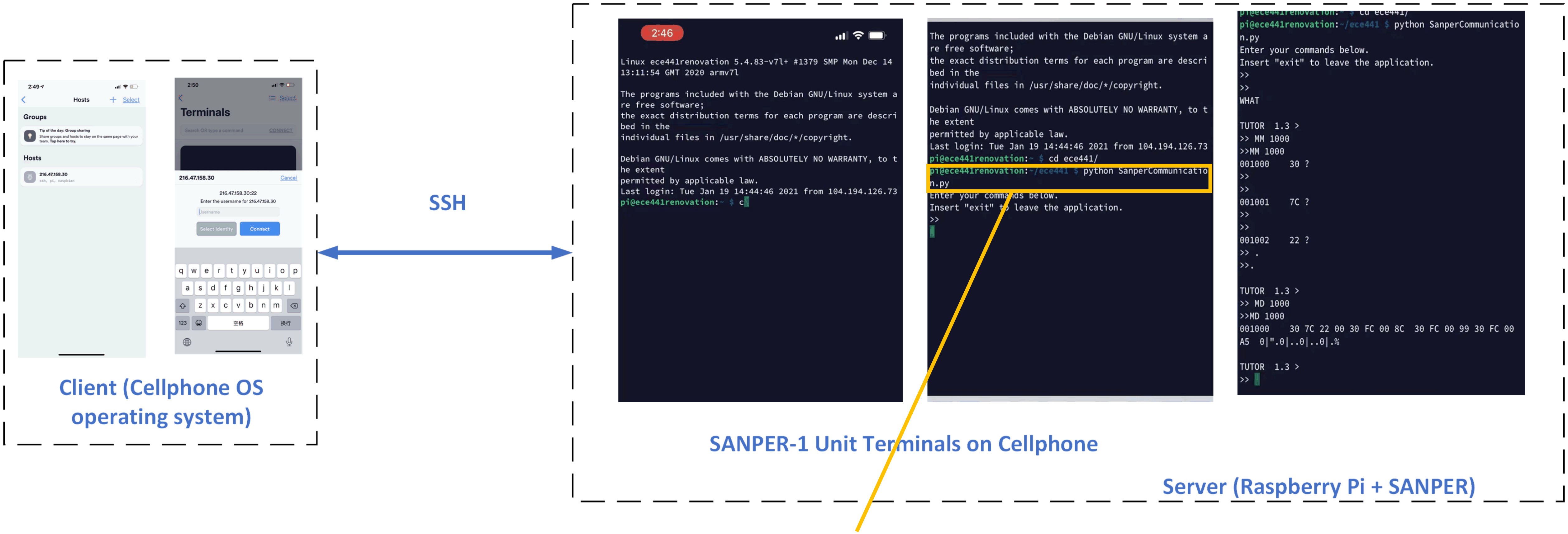
Signature: a4-b2-ab-ff-95-51-2e-91

```
pi@raspberrypi:~ $ cd ece441/
pi@raspberrypi:~/ece441 $ python SanperCommunication.py
SanperCommunication.py:12: RuntimeWarning: This channel is already in use, continuing anyway.  Use GPIO.s
etwarnings(False) to disable warnings.
  GPIO.setup(23,GPIO.OUT)
SanperCommunication.py:13: RuntimeWarning: This channel is already in use, continuing anyway.  Use GPIO.s
etwarnings(False) to disable warnings.
  GPIO.setup(24,GPIO.OUT)
Enter your commands below.
Insert "exit" to leave the application.
>> MM $2000
>>MM $2000
002000    AA ?
>> CD
>>CD
002001    4C ?
>> .
>>.

TUTOR  1.3 >
>> MM $2000
>>MM $2000
002000    CD ?
>> 
```

```
PHYSICAL ADDRESS=00000900
AAAA
```

```
AAAA
>> reset
SANPER has been reset.
>> reset
SANPER has been reset.
>> 
```

# Remote Access to SANPER using Smartphone



Tip of the day: Group sharing
Share groups and hosts to stay on the same page with your team. Tap here to try.

Enter the username for 216.47.158.30

Client (Cellphone OS operating system)

SSH

```
Linux ece441renovation 5.4.83-v7l+ #1379 SMP Mon Dec 14
13:11:54 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system a
re free software;
the exact distribution terms for each program are descri
bed in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to t
he extent
permitted by applicable law.
Last login: Tue Jan 19 14:44:46 2021 from 104.194.126.73
pi@ece441renovation:~ $ c
```

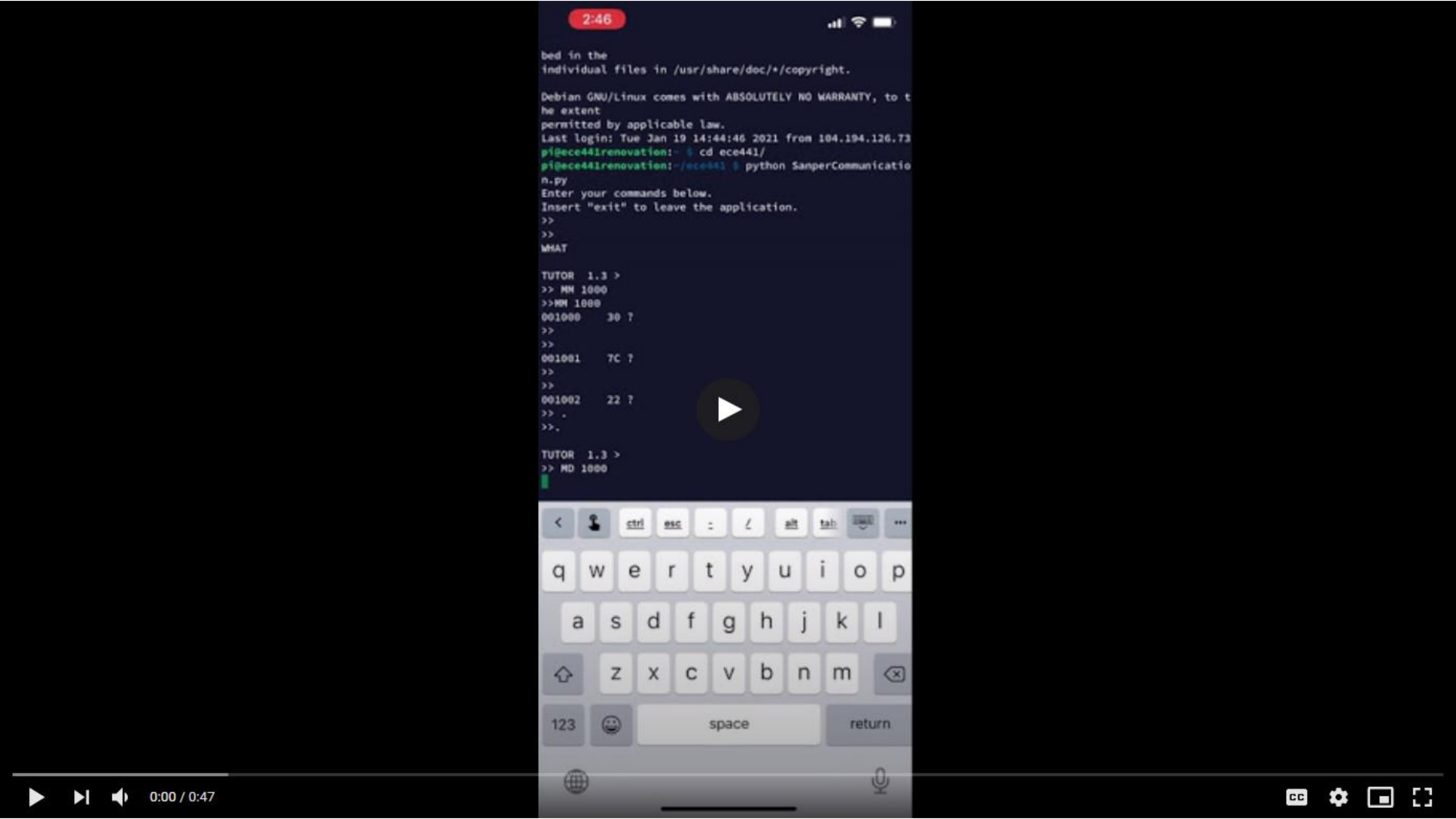
```
The programs included with the Debian GNU/Linux system a
re free software;
the exact distribution terms for each program are descri
bed in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to t
he extent
permitted by applicable law.
Last login: Tue Jan 19 14:44:46 2021 from 104.194.126.73
pi@ece441renovation:~ $ cd ece441/
pi@ece441renovation:~/ece441 $ python SanperCommunicatio
n.py
Enter your commands below.
Insert "exit" to leave the application.
>>
```

```
pi@ece441renovation:~ $ cd ece441/
pi@ece441renovation:~/ece441 $ python SanperCommunicatio
n.py
Enter your commands below.
Insert "exit" to leave the application.
>>
>>
WHAT

TUTOR  1.3 >
>> MM 1000
>>MM 1000
001000    30 ?
>>
>>
001001    7C ?
>>
>>
001002    22 ?
>> .
>>.

TUTOR  1.3 >
>> MD 1000
>>MD 1000
001000    30 7C 22 00 30 FC 00 8C  30 FC 00 99 30 FC 00
A5  0|".0|..0|..0|.%

TUTOR  1.3 >
>>
```

SANPER-1 Unit Terminals on Cellphone

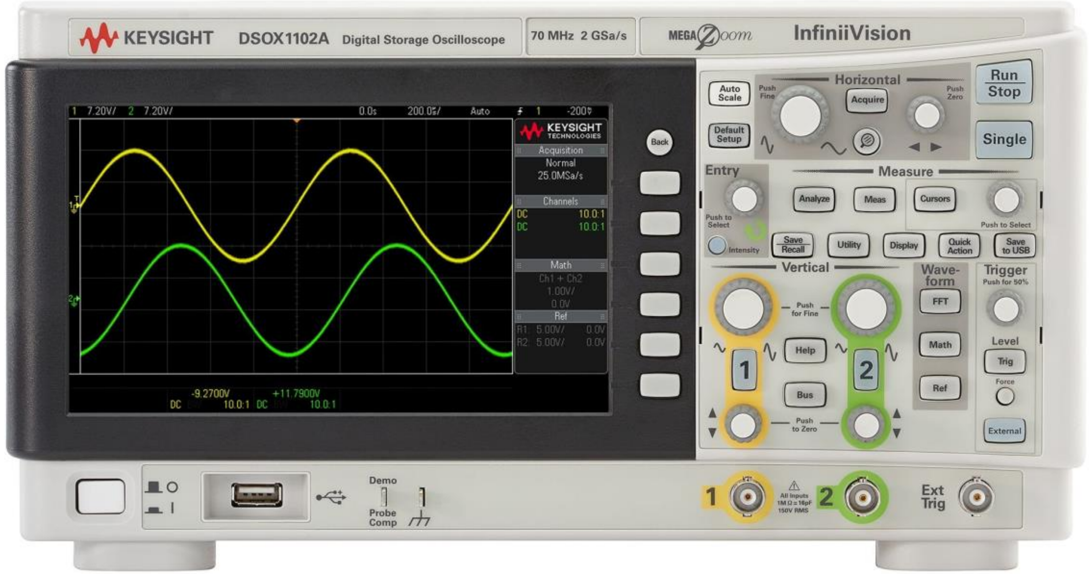
Server (Raspberry Pi + SANPER)

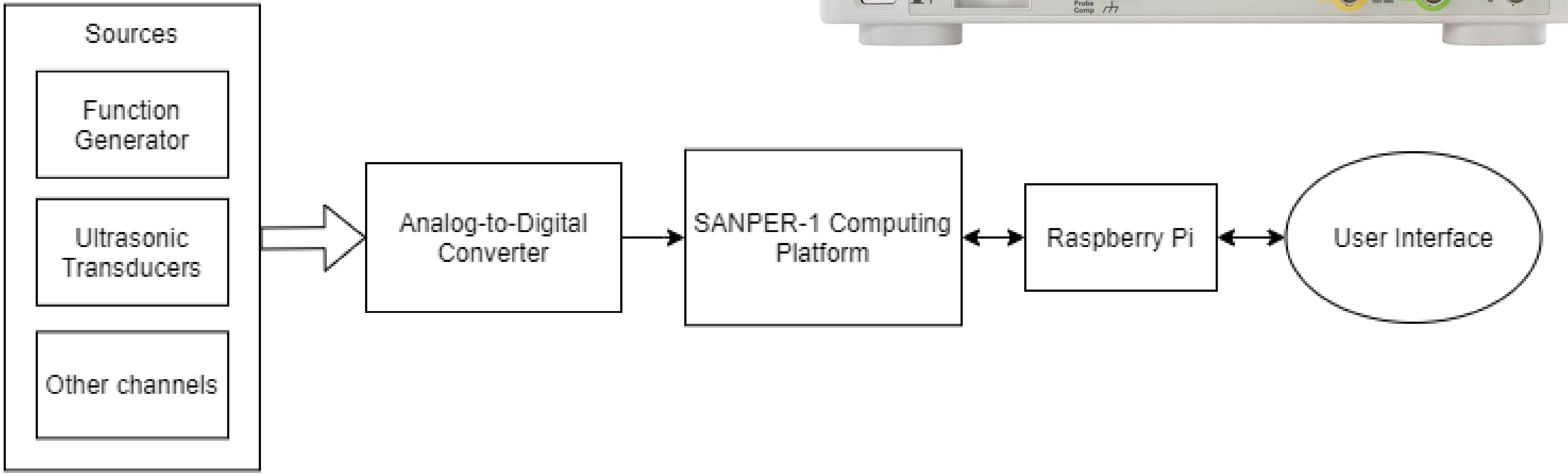
**The Python script to communicate with SANPER is called SanperCommuniation.py**



```
bed in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to t
he extent
permitted by applicable law.
Last login: Tue Jan 19 14:44:46 2021 from 104.194.126.73
pi@ece441renovation:~ $ cd ece441/
pi@ece441renovation:~/ece441 $ python SanperCommunicatio
n.py
Enter your commands below.
Insert "exit" to leave the application.
>>
>>
WHAT

TUTOR  1.3 >
>> MM 1000
>>MM 1000
001000    30 ?
>>
>>
001001    7C ?
>>
>>
001002    22 ?
>> .
>>.

TUTOR  1.3 >
>> MD 1000
```

# Digital Oscilloscope

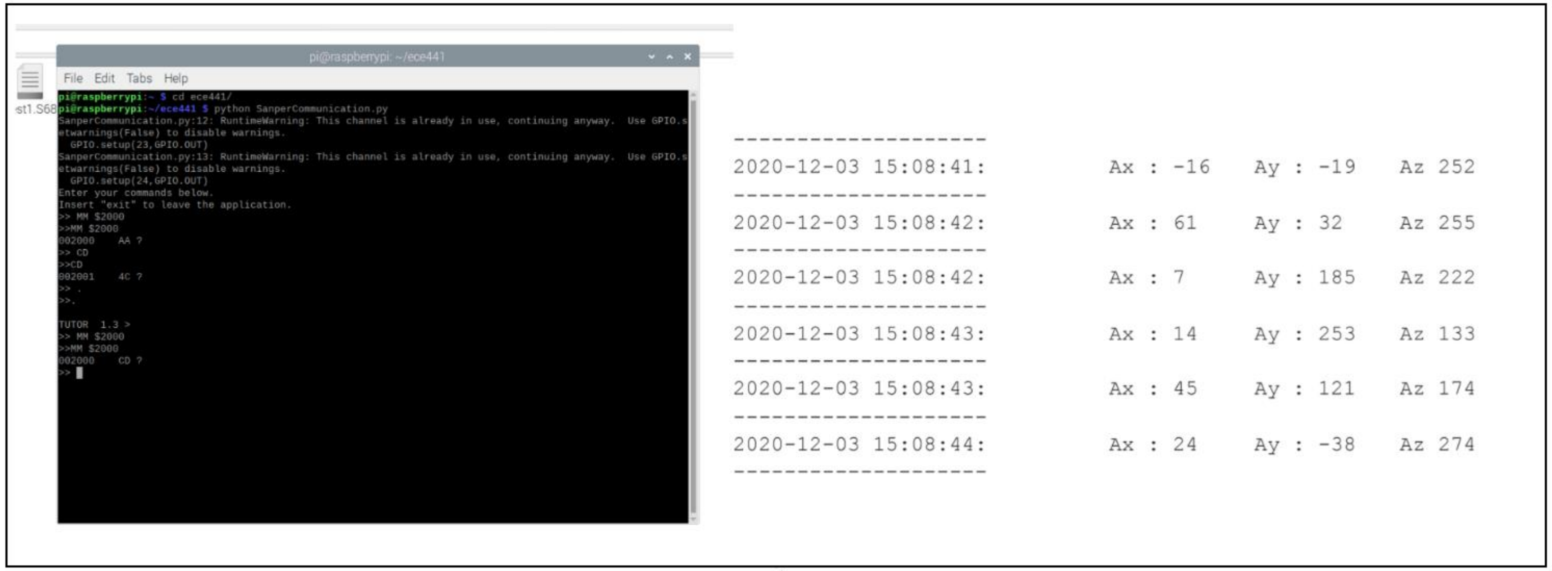




# Digital Oscilloscope

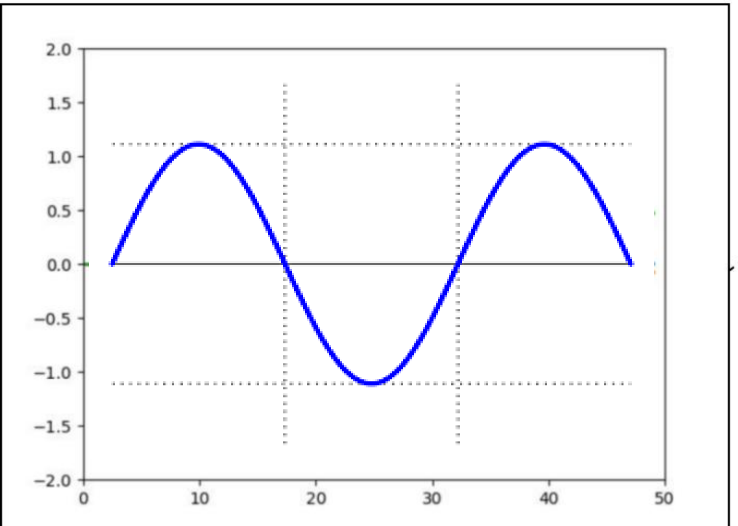
- This project is an oscilloscope that can pick up and display analog signals.
- A/D conversion is needed for interfacing with the signal inputs such as function generator or other analog signal sources.
- SANPER-1 Computing Platform transfer the sampled digital signals to the Raspberry Pi.
- **Hardware:**
  - **A/D Converter**
  - **Function Generator**

Python Script for Serial Communication, data decoding and SSH connection

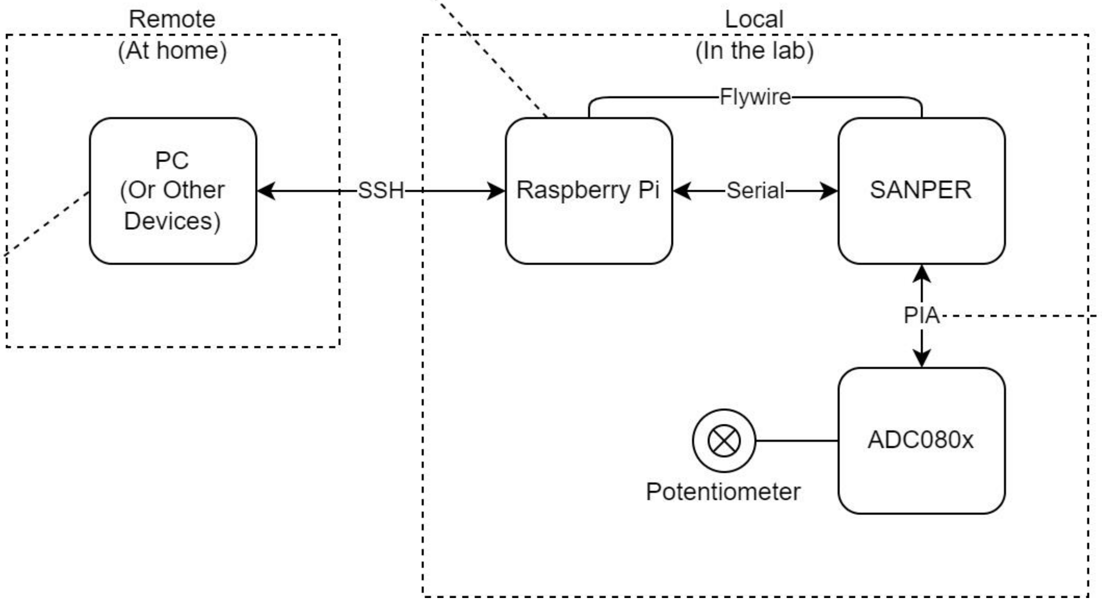


```
pi@raspberrypi:~ $ cd ece441/
pi@raspberrypi:~/ece441 $ python SanperCommunication.py
SanperCommunication.py:12: RuntimeWarning: This channel is already in use, continuing anyway.  Use GPIO.s
etwarnings(False) to disable warnings.
  GPIO.setup(23,GPIO.OUT)
SanperCommunication.py:13: RuntimeWarning: This channel is already in use, continuing anyway.  Use GPIO.s
etwarnings(False) to disable warnings.
  GPIO.setup(24,GPIO.OUT)
Enter your commands below.
Insert "exit" to leave the application.
>> MM $2000
>>MM $2000
002000    AA ?
>> CD
>>CD
002001    4C ?
>> .
>>.

TUTOR  1.3 >
>> MM $2000
>>MM $2000
002000    CD ?
>> 
```

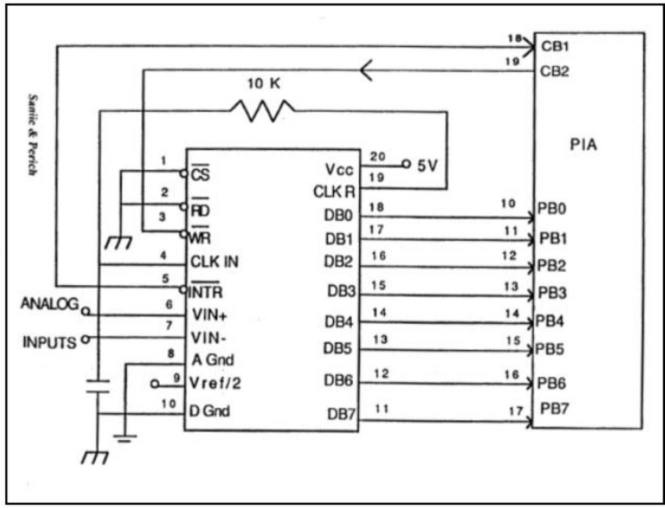
```
---------------------
2020-12-03 15:08:41:          Ax : -16    Ay : -19    Az 252
---------------------
2020-12-03 15:08:42:          Ax : 61     Ay : 32     Az 255
---------------------
2020-12-03 15:08:42:          Ax : 7      Ay : 185    Az 222
---------------------
2020-12-03 15:08:43:          Ax : 14     Ay : 253    Az 133
---------------------
2020-12-03 15:08:43:          Ax : 45     Ay : 121    Az 174
---------------------
2020-12-03 15:08:44:          Ax : 24     Ay : -38    Az 274
---------------------
```



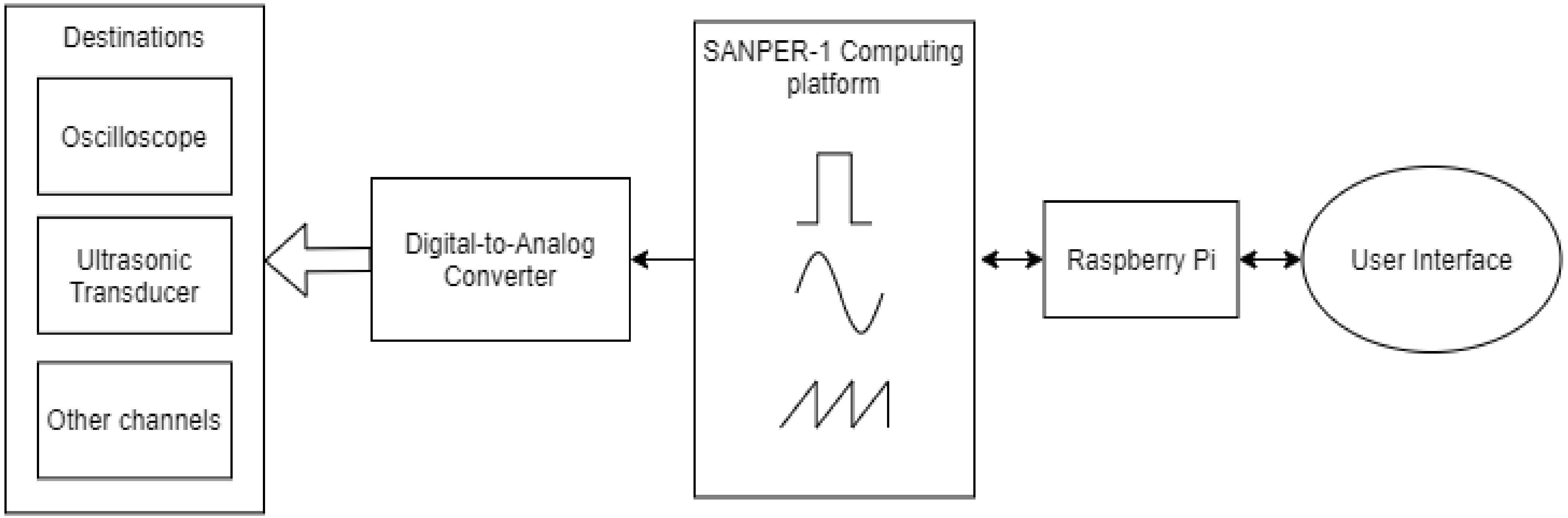
Graphic User Interface



Sample Design



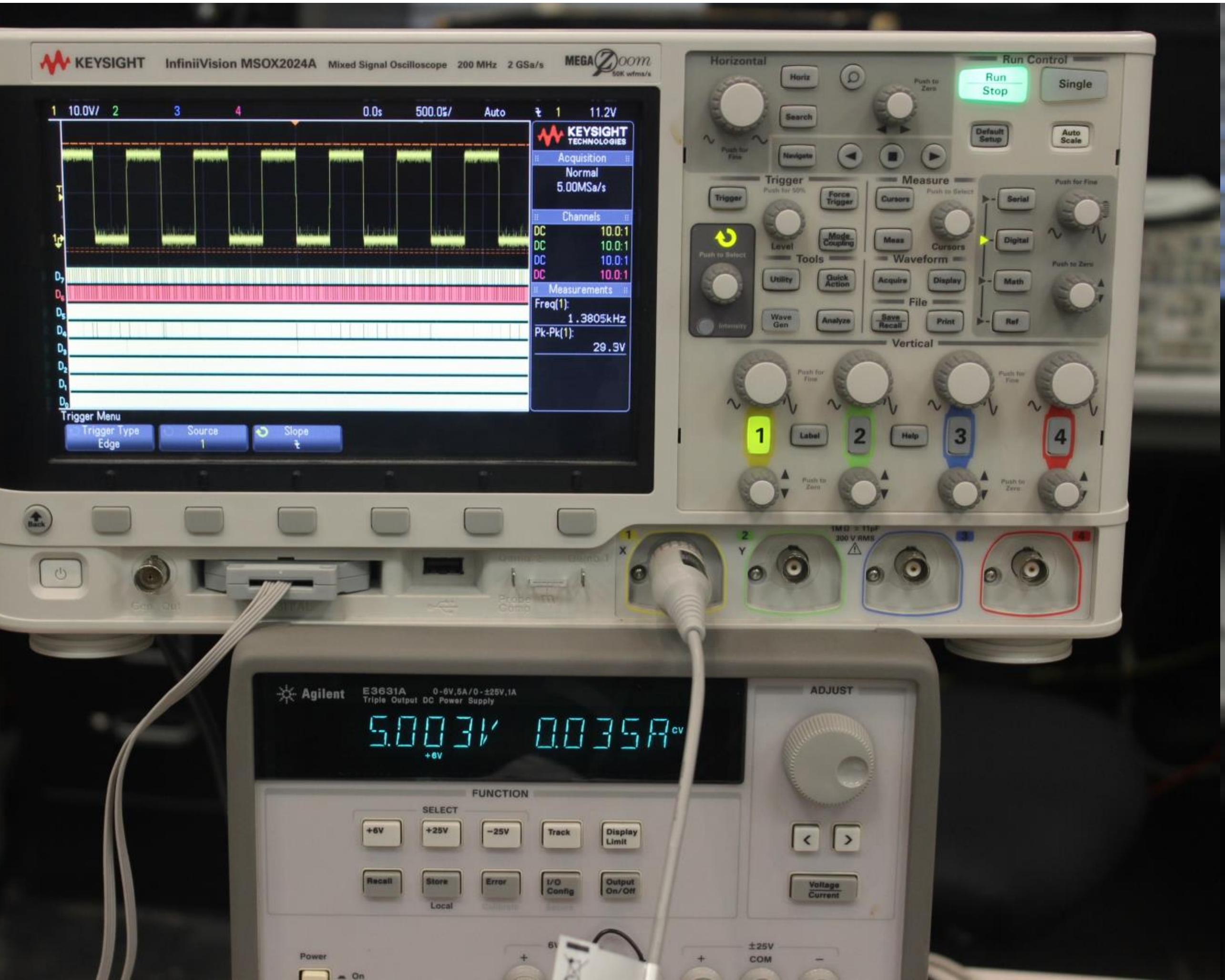
# Function Generator

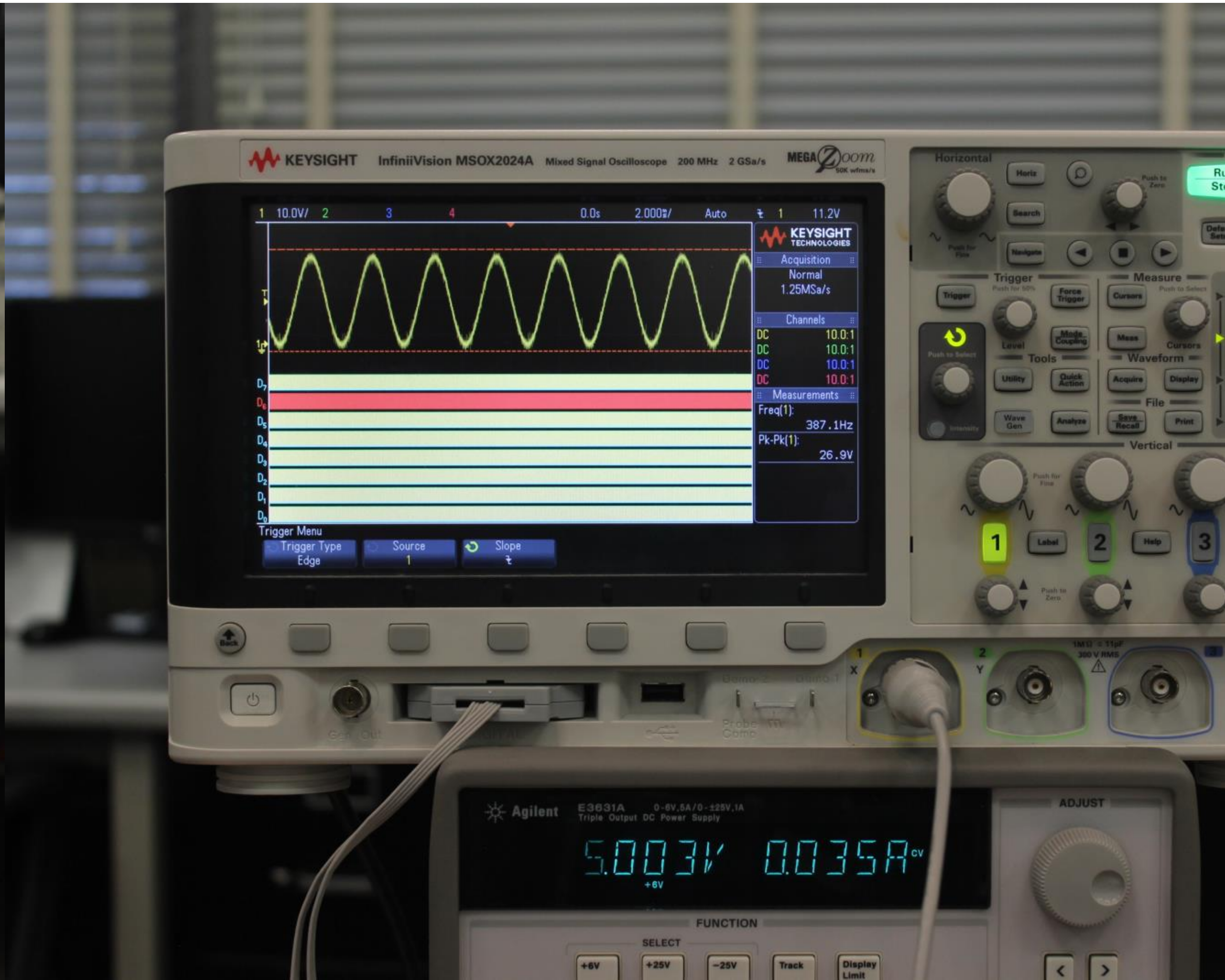


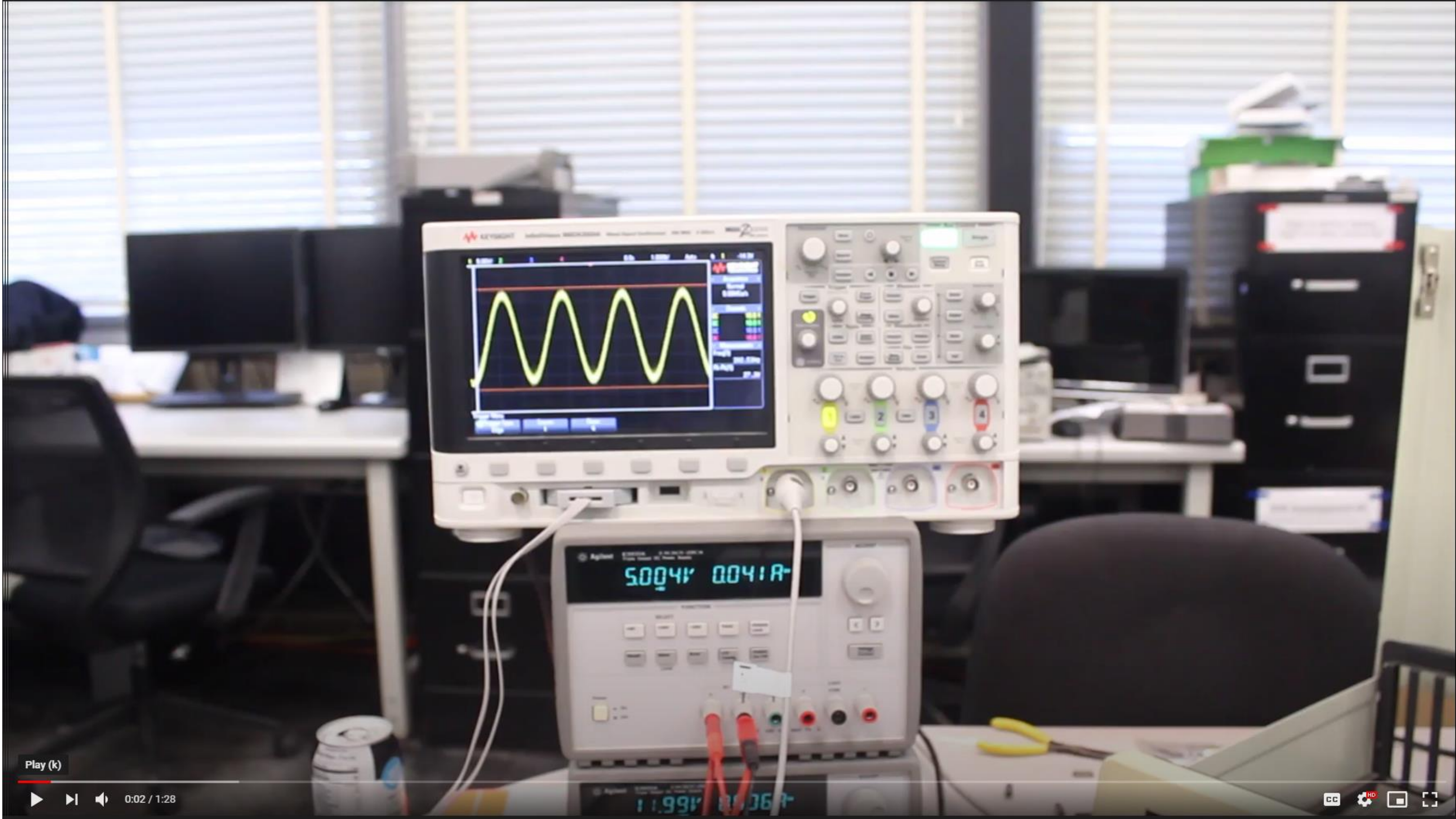
# Function Generator

- This project is a function generator based on the SANPER-1 Computing Platform.
- User interfaces are available for the transmitter.
- SANPER-1 provides the essential modulation and demodulation processes.
- A/D conversion and D/A conversion are needed for interfacing with the channels through the transducers.
- **Hardware:**
  - **A/D Converter**
  - **D/A Converter**
  - **Oscilloscope**

# Function Generator Results

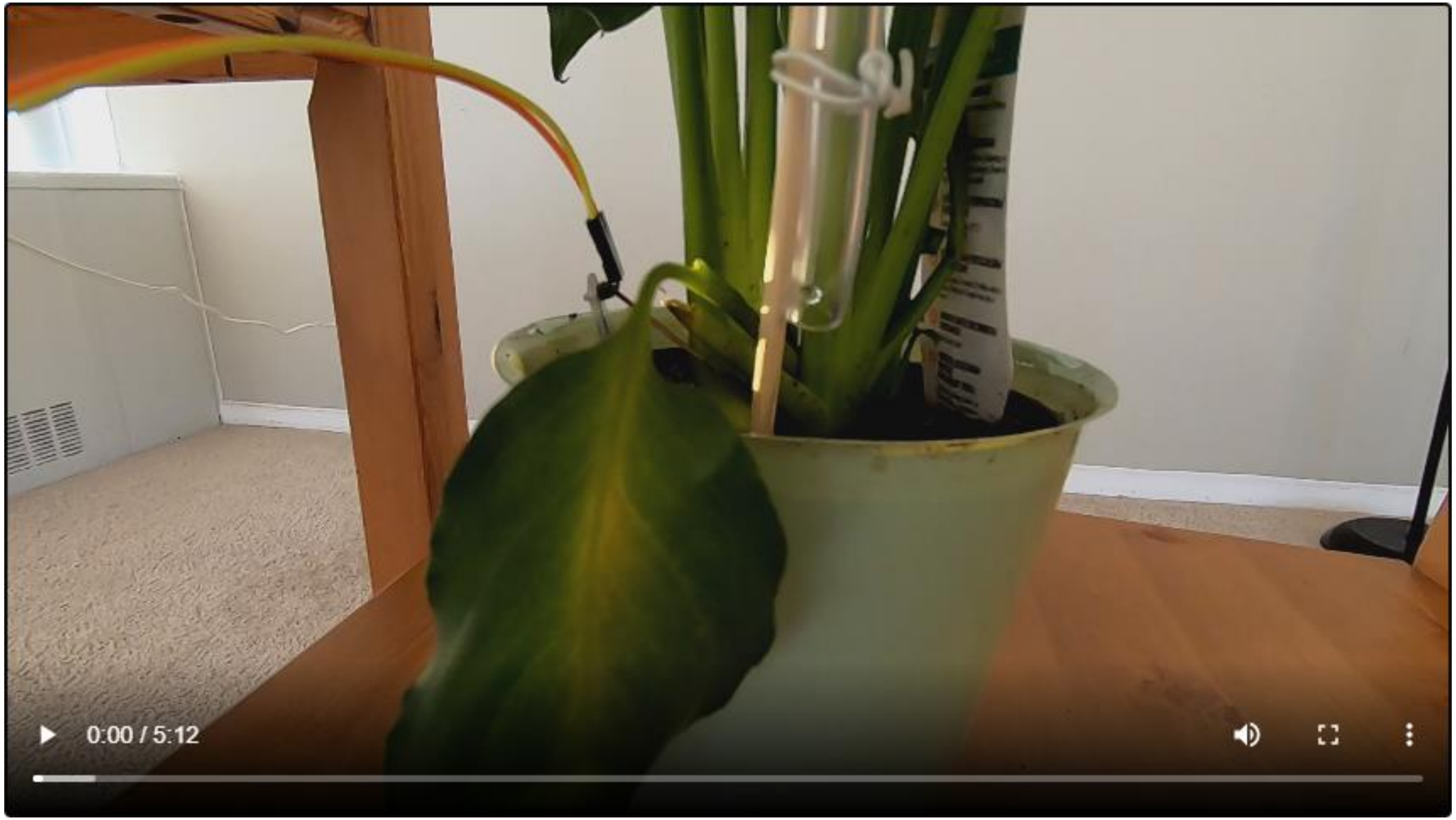





KEYSIGHT
Agilent
5.004V 0.041A
11.99V
Play (k)
0:02 / 1:28

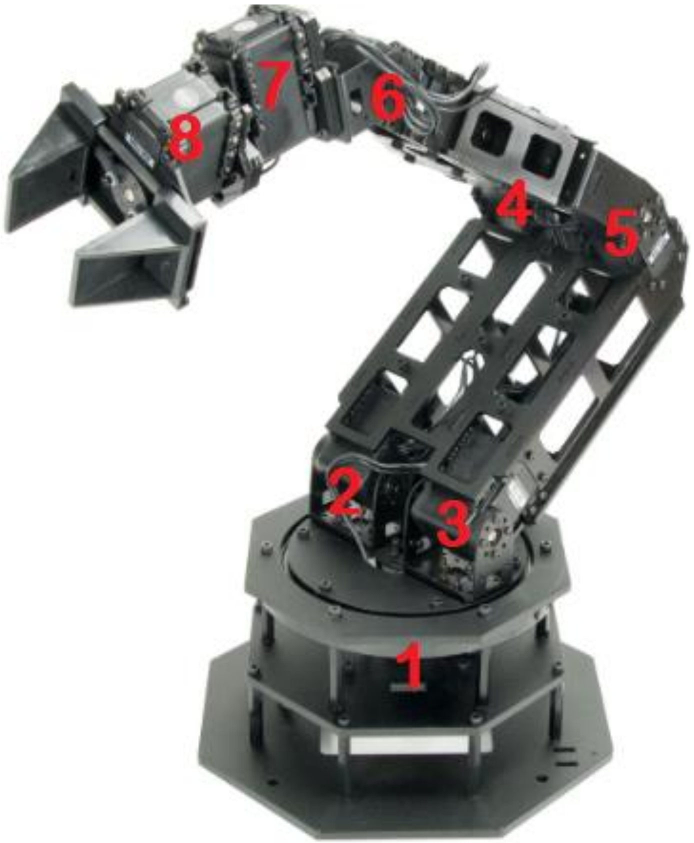
# Smart Irrigation System

- A smart irrigation system should continuously monitor the amount of water, determine if watering is required for the plants, supply the exact or approximate amount of water required for the plants, and lastly discontinue the water supply when the required amount has been delivered to the plants.
- It can be said that plant monitoring system is a part of smart irrigation. This project focuses on monitoring soil moisture, as well as temperature, humidity, and light intensity.


0:00 / 5:12

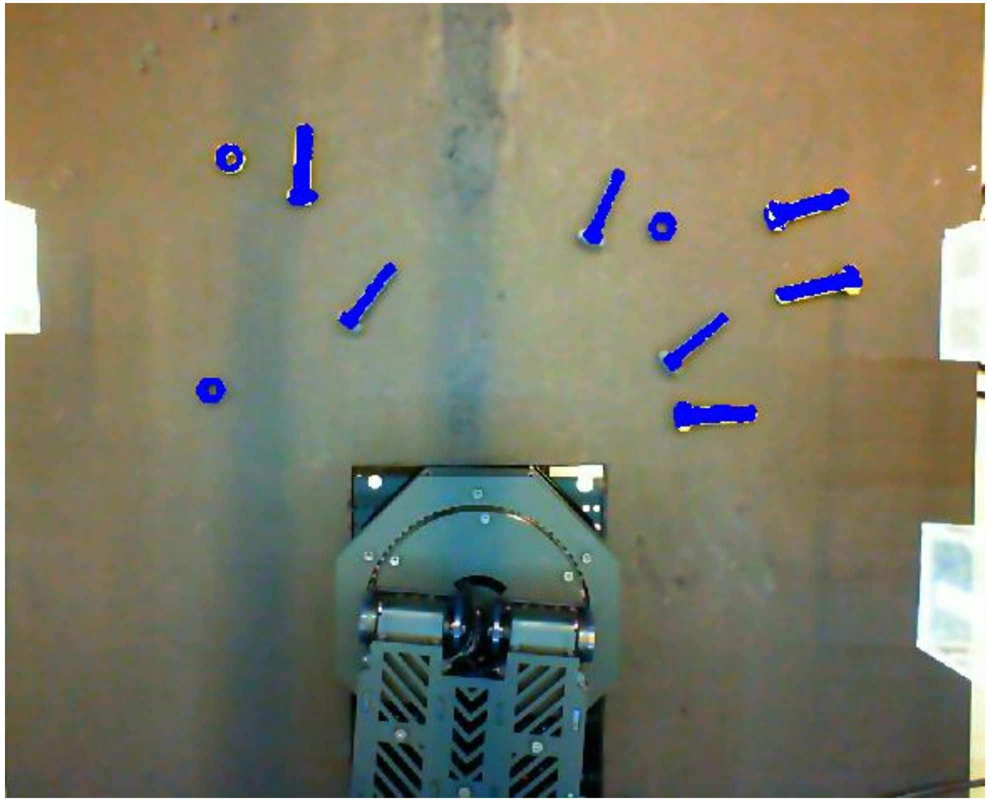
# Computer Vision Based Autonomous Robotic Arm

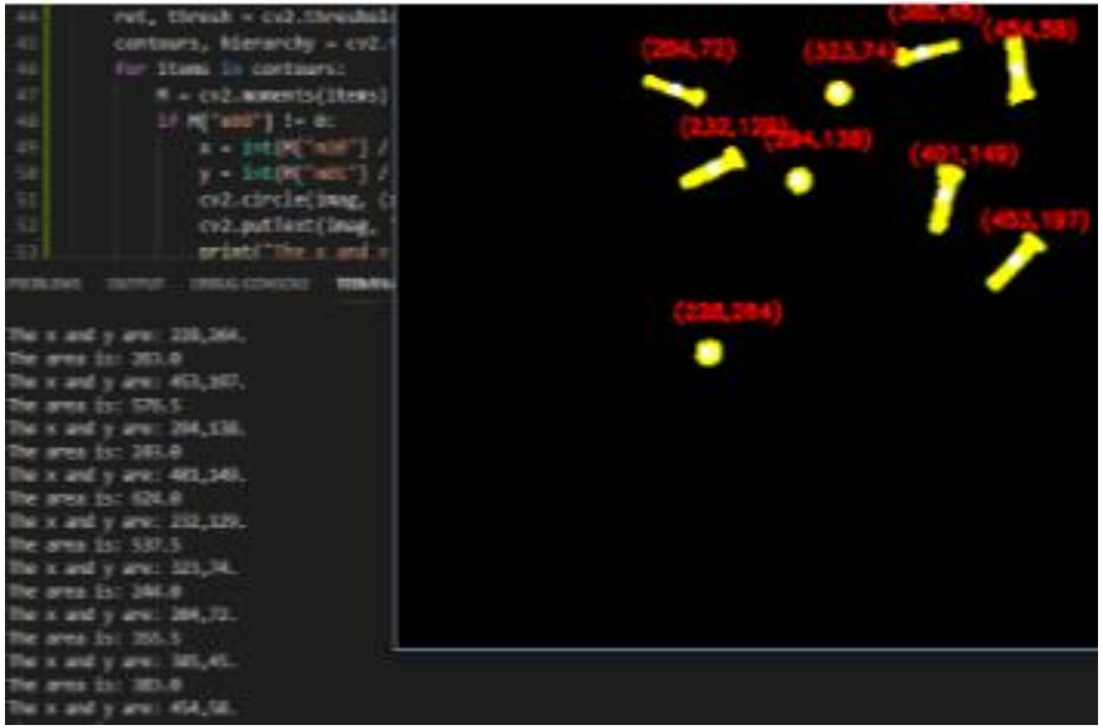
- The robotic arm will be used to identify, sort, and interact with objects within its reach.
- The objects that the robot will interact with will be different color nuts and bolts.
- A menu allows for the user to control the robot's speed, where it wants to go, what it should pick up, and even reverting an action.
- The red numbers on the picture marks the 8 motors that facilitate the move of the arm.



# Computer Vision Based Autonomous Robotic Arm

- The image is filtered, segmented and goes through an object recognition algorithm.





```
        ret, thresh = cv2.threshol
        contours, hierarchy = cv2.
        for items in contours:
            M = cv2.moments(items)
            if M['m00'] != 0:
                x = int(M['m10'] /
                y = int(M['m01'] /
                cv2.circle(img, (
                cv2.putText(img,
                print("The x and y
```

```
The x and y are: 228,264.
The area is: 283.0
The x and y are: 453,187.
The area is: 576.5
The x and y are: 294,138.
The area is: 283.0
The x and y are: 401,149.
The area is: 624.0
The x and y are: 232,129.
The area is: 537.5
The x and y are: 323,74.
The area is: 244.0
The x and y are: 204,72.
The area is: 355.5
The x and y are: 385,45.
The area is: 383.0
The x and y are: 454,58.
```

# Computer Vision Based Autonomous Robotic Arm Demo

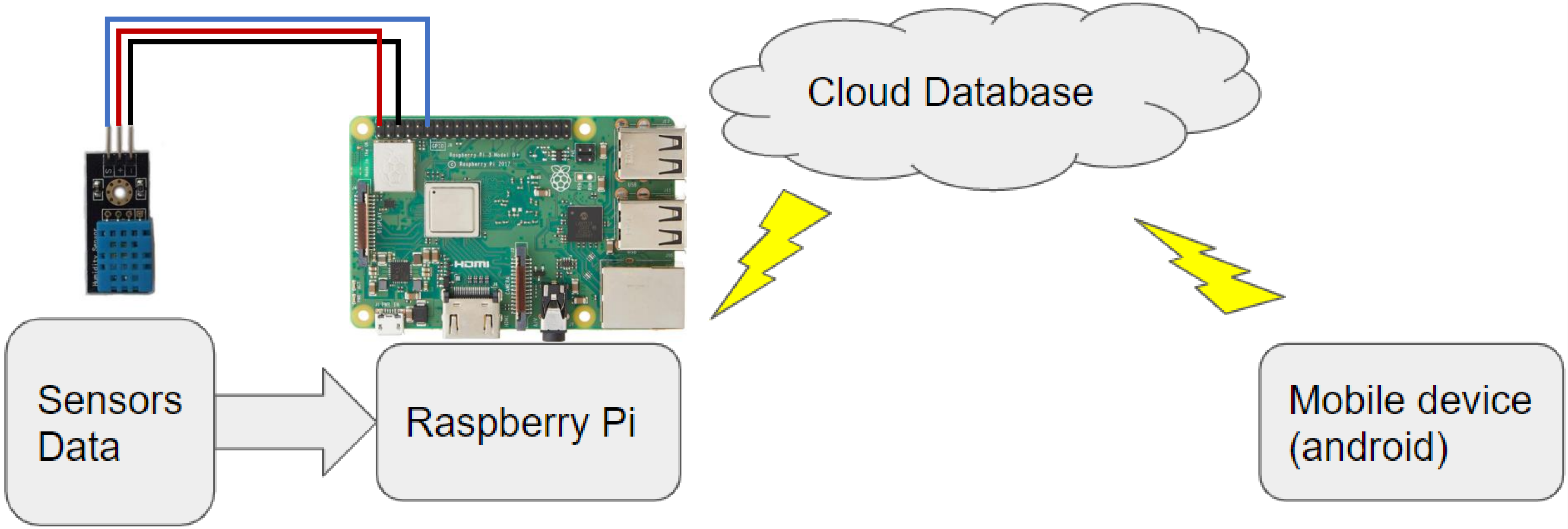
Embedded Computing and Signal Processing Research Laboratory

Department of Electrical and Computer Engineering

# Smartphone Application for Sensor Monitoring

- Sensor's data is acquired either from SANPER-1 or directly from sensors connected to the Raspberry Pi
- A python script runs on the Raspberry Pi uploads the data to the cloud every two seconds
- The android cell phone will update the displayed sensors data once there are changes to the cloud database

# Smartphone Application for Sensor Monitoring

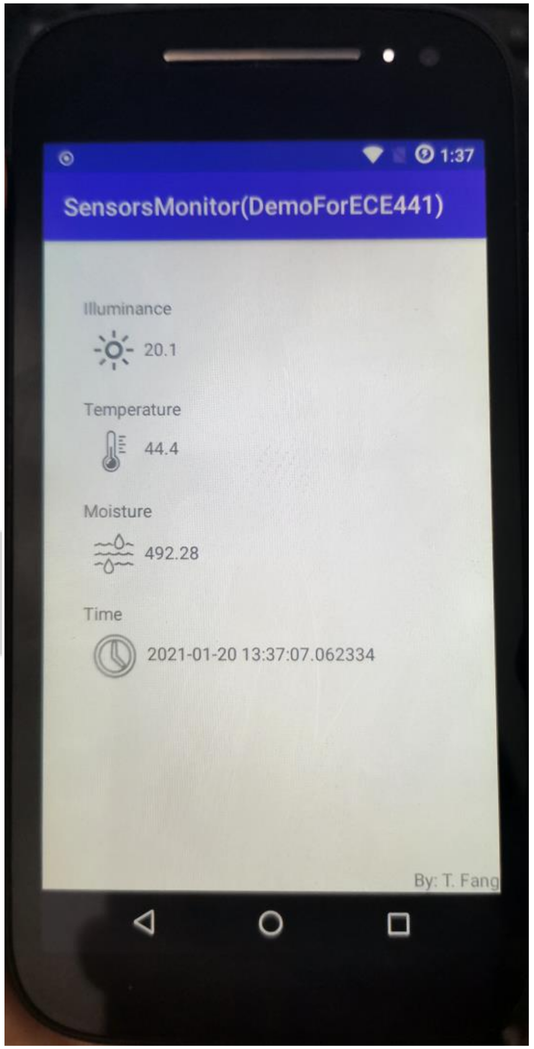


Cloud Database

Sensors Data

Raspberry Pi

Mobile device (android)



1:37

SensorsMonitor(DemoForECE441)

Illuminance

20.1

Temperature

44.4

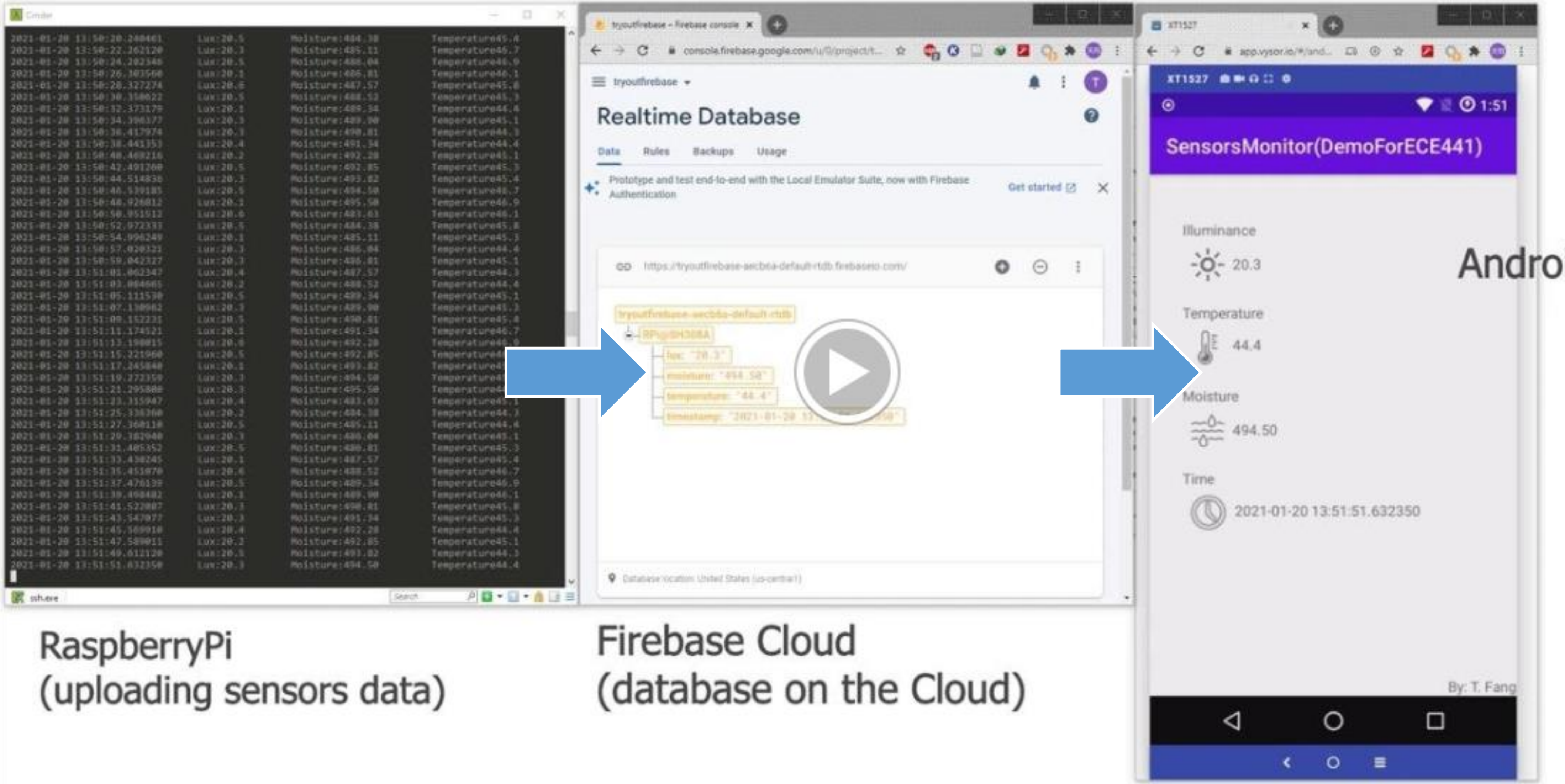
Moisture

492.28

Time

2021-01-20 13:37:07.062334

By: T. Fang



```
2021-01-20 13:50:20.240461     Lux:20.5     Moisture:484.38     Temperature45.4
2021-01-20 13:50:22.262120     Lux:20.3     Moisture:485.11     Temperature46.7
2021-01-20 13:50:24.282346     Lux:20.5     Moisture:486.04     Temperature46.9
2021-01-20 13:50:26.303560     Lux:20.1     Moisture:486.81     Temperature46.1
2021-01-20 13:50:28.327274     Lux:20.6     Moisture:487.57     Temperature45.8
2021-01-20 13:50:30.350022     Lux:20.5     Moisture:488.52     Temperature45.3
2021-01-20 13:50:32.373179     Lux:20.1     Moisture:489.34     Temperature44.4
2021-01-20 13:50:34.396377     Lux:20.3     Moisture:489.90     Temperature45.1
2021-01-20 13:50:36.417974     Lux:20.3     Moisture:490.81     Temperature44.3
2021-01-20 13:50:38.441353     Lux:20.4     Moisture:491.34     Temperature44.4
2021-01-20 13:50:40.469216     Lux:20.2     Moisture:492.28     Temperature45.1
2021-01-20 13:50:42.491260     Lux:20.5     Moisture:492.85     Temperature45.3
2021-01-20 13:50:44.514836     Lux:20.3     Moisture:493.82     Temperature45.4
2021-01-20 13:50:46.539185     Lux:20.5     Moisture:494.50     Temperature46.7
2021-01-20 13:50:48.926812     Lux:20.1     Moisture:495.50     Temperature46.9
2021-01-20 13:50:50.951512     Lux:20.6     Moisture:483.63     Temperature46.1
2021-01-20 13:50:52.972333     Lux:20.5     Moisture:484.38     Temperature45.8
2021-01-20 13:50:54.996249     Lux:20.1     Moisture:485.11     Temperature45.3
2021-01-20 13:50:57.020321     Lux:20.3     Moisture:486.04     Temperature44.4
2021-01-20 13:50:59.042327     Lux:20.3     Moisture:486.81     Temperature45.1
2021-01-20 13:51:01.062347     Lux:20.4     Moisture:487.57     Temperature44.3
2021-01-20 13:51:03.084665     Lux:20.2     Moisture:488.52     Temperature44.4
2021-01-20 13:51:05.111530     Lux:20.5     Moisture:489.34     Temperature45.1
2021-01-20 13:51:07.130962     Lux:20.3     Moisture:489.90     Temperature45.3
2021-01-20 13:51:09.152231     Lux:20.5     Moisture:490.81     Temperature45.4
2021-01-20 13:51:11.174521     Lux:20.1     Moisture:491.34     Temperature46.7
2021-01-20 13:51:13.190015     Lux:20.6     Moisture:492.28     Temperature46.9
2021-01-20 13:51:15.221960     Lux:20.5     Moisture:492.85     Temperature4
2021-01-20 13:51:17.245840     Lux:20.1     Moisture:493.82     Temperature4
2021-01-20 13:51:19.272359     Lux:20.3     Moisture:494.50     Temperature4
2021-01-20 13:51:21.295808     Lux:20.3     Moisture:495.50     Temperature4
2021-01-20 13:51:23.315947     Lux:20.4     Moisture:483.63     Temperature45.1
2021-01-20 13:51:25.338360     Lux:20.2     Moisture:484.38     Temperature44.3
2021-01-20 13:51:27.360110     Lux:20.5     Moisture:485.11     Temperature44.4
2021-01-20 13:51:29.382040     Lux:20.3     Moisture:486.04     Temperature45.1
2021-01-20 13:51:31.405352     Lux:20.5     Moisture:486.81     Temperature45.3
2021-01-20 13:51:33.430245     Lux:20.1     Moisture:487.57     Temperature45.4
2021-01-20 13:51:35.451070     Lux:20.6     Moisture:488.52     Temperature46.7
2021-01-20 13:51:37.476139     Lux:20.5     Moisture:489.34     Temperature46.9
2021-01-20 13:51:39.498482     Lux:20.1     Moisture:489.90     Temperature46.1
2021-01-20 13:51:41.522007     Lux:20.3     Moisture:490.81     Temperature45.8
2021-01-20 13:51:43.547077     Lux:20.3     Moisture:491.34     Temperature45.3
2021-01-20 13:51:45.569010     Lux:20.4     Moisture:492.28     Temperature44.4
2021-01-20 13:51:47.589011     Lux:20.2     Moisture:492.85     Temperature45.1
2021-01-20 13:51:49.612120     Lux:20.5     Moisture:493.82     Temperature44.3
2021-01-20 13:51:51.632350     Lux:20.3     Moisture:494.50     Temperature44.4
```

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

https://tryoutfirebase-aecb6a-default-rtdb.firebaseio.com/

```
tryoutfirebase-aecb6a-default-rtdb
 └ RPi@SH308A
    ├ lux: "20.3"
    ├ moisture: "494.50"
    ├ temperature: "44.4"
    └ timestamp: "2021-01-20 13:...50"
```

Database location: United States (us-central1)

SensorsMonitor(DemoForECE441)

Illuminance
20.3

Temperature
44.4

Moisture
494.50

Time
2021-01-20 13:51:51.632350

By: T. Fang

RaspberryPi
(uploading sensors data)

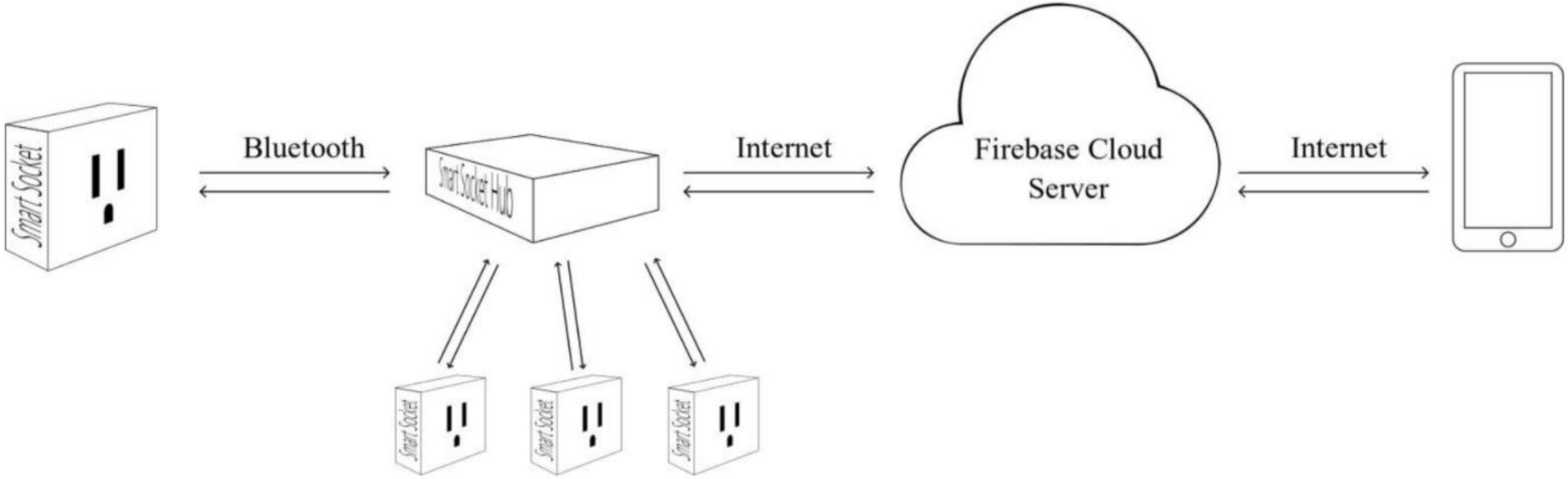
Firebase Cloud
(database on the Cloud)

Android App

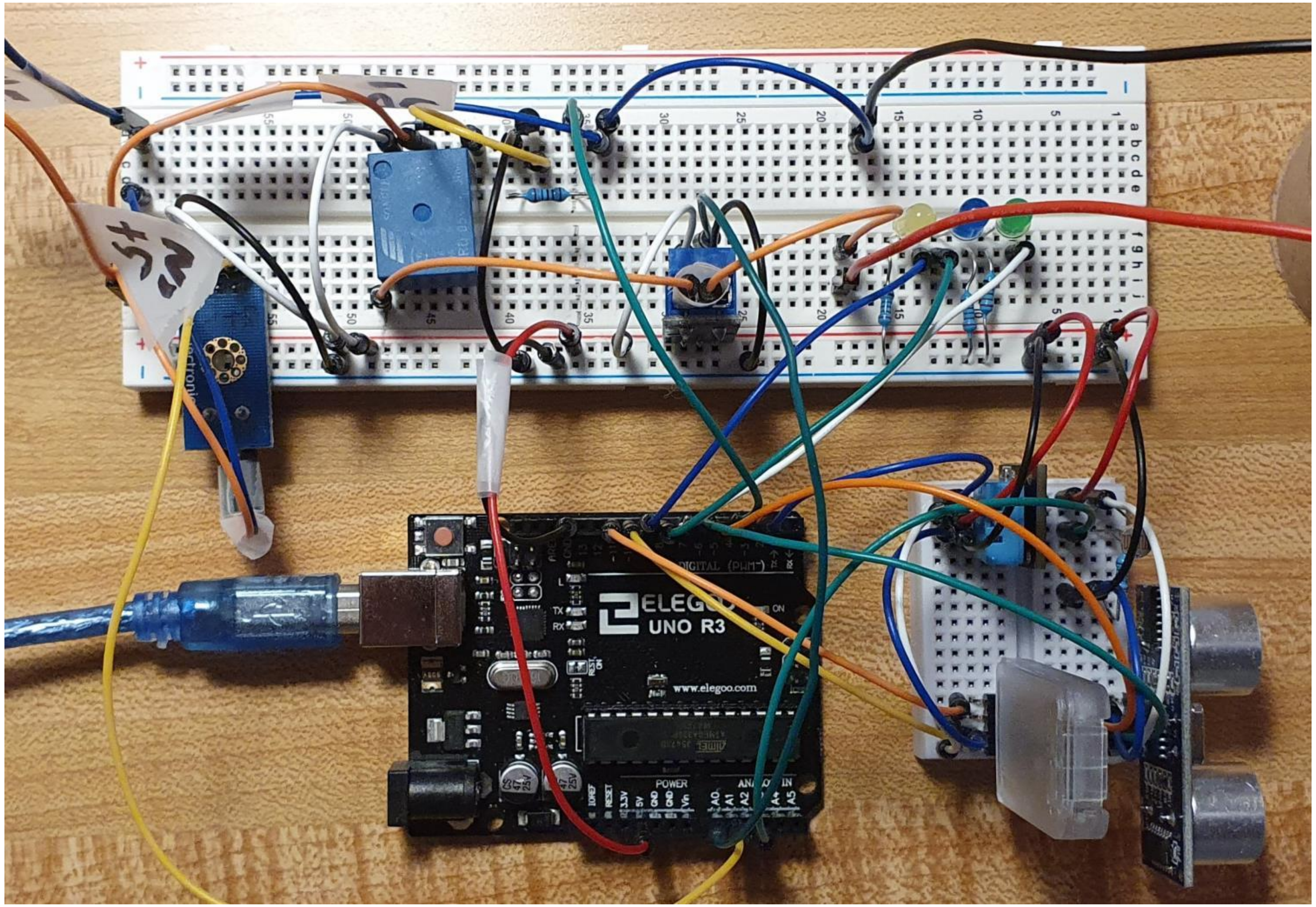
# Smart Socket

- The aim of this project is to build a custom ‘Smart Socket’ (SS) based on Arduino UNO and Raspberry Pi 3.
- Its main function is to provide services such as remote control and monitoring, reliable electrical protection and tasks automation.
- The remote control of the socket is based on a mobile application, which uses cloud services and allows the user to perform all tasks remotely.
- When combining the ‘Smart Socket’ with household appliances that are not connected or smart by design, they will instantly become active elements for the user, offering new and smart functionalities to optimize and improve its usage.

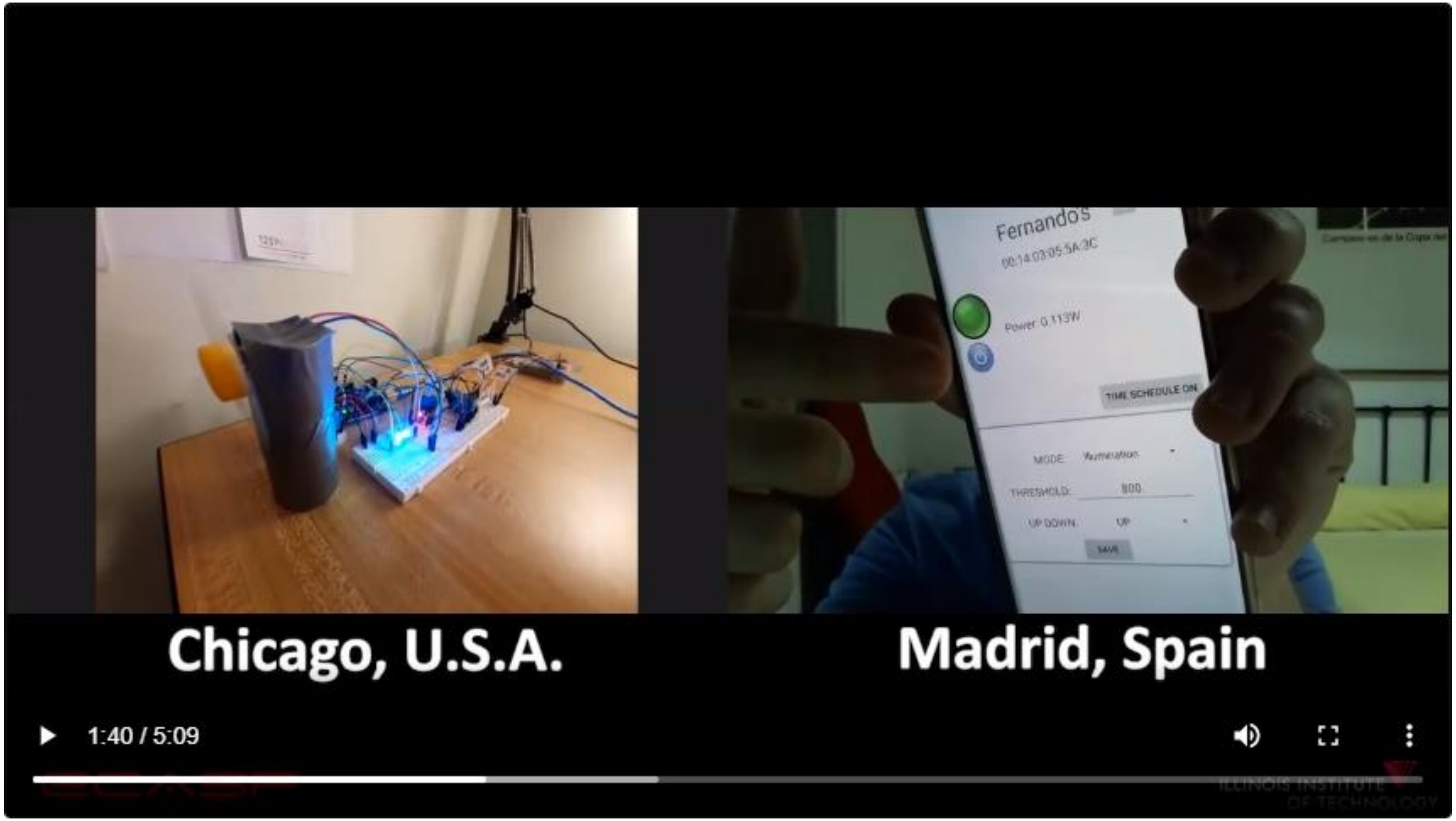
# Smart Socket General Architecture



# Smart Socket



# Smart Socket Demo:
# Remote Control Lights Across The Atlantic



Fernando's

00:14:03:05:5A:3C

Power: 0.113W

MODE: Illumination

THRESHOLD: 800

UP DOWN: UP

# Thank You