

DAB+ Labs สำหรับ Raspberry Pi

คู่มือการเรียนรู้ Digital Audio Broadcasting Plus

พร้อม RTL-SDR และ PyQt5

เวอร์ชัน 1.1 | กันยายน 2025


วัตถุประสงค์โครงการ

เรียนรู้เทคโนโลยี

- DAB+ จากพื้นฐานจนถึงขั้นสูง
- Python & PyQt5 GUI programming
- Software Defined Radio (SDR)
- RF Signal Processing

สร้างแอปพลิเคชัน

- DAB+ Station Scanner
- Program Recorder
- Signal Analyzer
- Touch-Friendly GUI

 เป้าหมาย: สร้างแอปที่ใช้งานได้จริงบน Raspberry Pi

ข้อกำหนดระบบ

Hardware

- Raspberry Pi 4 (4GB+ RAM)
- RTL-SDR V4 Dongle
- หน้าจอสัมผัส 7" (HDMI)
- หูฟัง 3.5mm
- เส้าอากาศ DAB+

Software

- Raspberry Pi OS Bookworm
- Python 3.11+
- PyQt5 GUI Framework
- welle.io DAB+ Decoder
- RTL-SDR Libraries

ภาพรวมแล็บทั้งหมด

Lab Series พื้นฐาน (เดิม - เน้น tools)

Lab	หัวข้อ	เวลา	ระดับ
1	การตั้งค่าเบื้องต้น RTL-SDR	20 นาที	★ ★
2	การรับสัญญาณ DAB+ พื้นฐาน	25 นาที	★ ★ ★
3	Command Line Tools สำหรับ DAB+	15 นาที	★ ★ ★
4	ETISnoop Analysis	15 นาที	★ ★ ★ ★
5	สรุปและวิเคราะห์ขั้นสูง	10 นาที	★ ★ ★ ★ ★

รวมเวลา: ~1.25 ชั่วโมง (75 นาที) |  เป้าหมาย: เรียนรู้การใช้งาน DAB+ tools

Lab Series ขั้นสูง (ใหม่ - เน้น development + trap exercises)

Lab	หัวข้อ	เวลา	ระดับ
-----	--------	------	-------

LAB 0: Introduction to DAB+, Python และ PyQt5

 เวลารวม: 75 นาที (1 ชั่วโมง 15 นาที)

 ภาพรวมเนื้อหา

เป็นแล็บพื้นฐานสำหรับมือใหม่ ที่ยังไม่เคยใช้ Python หรือไม่รู้จัก DAB+

ส่วนที่ 1: DAB+ Technology (15 นาที)

DAB+ vs FM Radio

- เสียงดิจิทัล ไม่มี static หรือสัญญาณรบกวน
- คุณภาพคงที่ ไม่ขึ้นกับระยะทาง
- Metadata ชื่อเพลง, ศิลปิน แบบ real-time
- MOT Slideshow รูปภาพ album art
- Multiplexing หลายสถานีใช้ความถี่เดียว

DAB+ ในประเทศไทย (2025)

ความถี่ตาม NBTC:

- **Channel 5C:** 178.352 MHz (Bangkok, Pattaya, Hua Hin)
- **Channel 6C:** 185.360 MHz (National Network)
- **Channel 7C:** 192.352 MHz (เชียงใหม่, ภาคใต้)
- **Channel 8C:** 199.360 MHz (Bangkok, Pattaya, Hua Hin)

🐍 ส่วนที่ 2: Python สำหรับมือใหม่ (30 นาที)

Python Basics

```
# Variables และ Data Types
name = "สวัสดี"      # String
age = 25              # Integer
height = 175.5        # Float
is_student = True     # Boolean

# Lists และการใช้งาน
fruits = ["แอปเปิ้ล", "กล้วย", "ส้ม"]
fruits.append("มะม่วง")
print(len(fruits))    # แสดง: 4
```

Control Flow

```
# Loops (การวนซ้ำ)
for fruit in fruits:
    print("ผลไม้:", fruit)

# Conditions (เงื่อนไข)
if age >= 18:
    print("เป็นผู้ใหญ่แล้ว")
else:
    print("เป็นเด็ก")

# Functions (ฟังก์ชัน)
def say_hello(name):
    return "สวัสดี " + name
```


Python: Classes และ Hardware Integration

Object-Oriented Programming

```
class DABStation:
    def __init__(self, name, frequency):
        self.name = name
        self.frequency = frequency
        self.is_playing = False

    def start_playing(self):
        self.is_playing = True
        print(f"เริ่มเล่น {self.name}")

    def stop_playing(self):
        self.is_playing = False
```

Raspberry Pi GPIO

```
try:
    import RPi.GPIO as GPIO
    import time

    # ตั้งค่า GPIO pin 18
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(18, GPIO.OUT)

    # กระพริบ LED
    GPIO.output(18, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(18, GPIO.LOW)

except ImportError:
    print("ทำงานบนคอมพิวเตอร์ทั่วไป")
```

ส่วนที่ 3: PyQt5 Hands-on (30 นาที)

PyQt5 Components

```
from PyQt5.QtWidgets import *
import sys

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setup_ui()

    def setup_ui(self):
        # สร้าง central widget
        central_widget = QWidget()
        self.setCentralWidget(central_widget)

        # สร้าง layout
        layout = QVBoxLayout(central_widget)
```

Touch-Friendly Design

```
# ปุ่มขนาดใหญ่สำหรับสัมผัส
button = QPushButton("กดที่นี่")
button.setMinimumSize(120, 60)

# Font ขนาดใหญ่
font = QFont()
font.setPointSize(14)
button.setFont(font)

# CSS Styling
button.setStyleSheet("""
    QPushButton {
        border-radius: 8px;
        background: #3498db;
        color: white;
```



PyQt5: Signals & Slots

Event Handling

```
class DABPlayerWidget(QWidget):
    def __init__(self):
        super().__init__()
        self.setup_ui()
        self.setup_connections()

    def setup_connections(self):
        # เชื่อม signals กับ slots
        self.play_button.clicked.connect(self.on_play)
        self.volume_slider.valueChanged.connect(self.on_volume_change)

    def on_play(self):
        print("เริ่มเล่นเพลง!")
```



QTimer และ Updates

```
from PyQt5.QtCore import QTimer

class SignalMonitor(QWidget):
    def __init__(self):
        super().__init__()

        # Timer สำหรับ real-time update
        self.timer = QTimer()
        self.timer.timeout.connect(self.update_signal)
        self.timer.start(1000) # อัปเดตทุก 1 วินาที

    def update_signal(self):
        # อัปเดตค่าสัญญาณ
        signal_strength = self.get_signal_strength()
        self.signal_bar.setValue(signal_strength)
```

LAB 0: Demo Applications

Demo 1: Basic Widgets

- QLabel แสดงข้อความและรูปภาพ
- QPushButton ปุ่มกดขนาดใหญ่
- QLineEdit ช่องใส่ข้อความ
- QTextEdit พื้นที่ข้อความหลายบรรทัด
- QSlider แถบเลื่อนค่า
- QProgressBar แสดงความคืบหน้า

สิ่งที่นักเรียนต้องเติม:

1. อ่านชื่อ จาก QLineEdit และแสดงใน QLabel
2. ควบคุม QProgressBar ด้วย QSlider
3. เริ่ม/หยุด QTimer และแสดงเวลาปัจจุบัน

Demo 2: Touch Interface

- ขนาดปุ่ม อย่างน้อย 60x40 pixels
- Font Size 12-16pt สำหรับหน้าจอ 7"
- Visual Feedback เปลี่ยนสีเมื่อกด
- Layout Management responsive design
- Error Handling การจัดการข้อผิดพลาด

LAB 0: ผลลัพธ์ที่คาดหวัง

ความรู้ที่ได้รับ

DAB+ Technology:

- เข้าใจความแตกต่างจาก FM
- รู้จักเทคโนโลยี RTL-SDR
- เข้าใจ DAB+ ในประเทศไทย

Python Programming:

- Variables, functions, classes
- File handling และ modules
- GPIO programming พื้นฐาน

Skills ที่พร้อมใช้

PyQt5 GUI Development:

- Widget การใช้งานพื้นฐาน
- Signals & Slots system
- Touch-friendly UI design
- Real-time updates ด้วย QTimer

เตรียมพร้อม สำหรับ Labs ขั้นสูง!

LAB 1: การตั้งค่าเบื้องต้น RTL-SDR (20 นาที)

การติดตั้ง RTL-SDR Drivers

```
# อัปเดตระบบ
sudo apt update && sudo apt upgrade -y

# ติดตั้ง dependencies
sudo apt install libusb-1.0-0-dev git cmake pkg-config build-essential

# ดาวน์โหลดและติดตั้ง RTL-SDR Blog drivers
git clone https://github.com/rtlsdrblog/rtl-sdr-blog
cd rtl-sdr-blog
mkdir build && cd build
cmake ../ -DINSTALL_UDEV_RULES=ON
make
sudo make install
sudo ldconfig

# Blacklist DVB-T drivers
echo 'blacklist dvb_usb_rtl28xxu' | sudo tee --append /etc/modprobe.d/blacklist-dvb_usb_rtl28xxu.conf
```

การทดสอบ

```
# รีบูตเครื่อง
sudo reboot

# ทดสอบ RTL-SDR
rtl_test -t
```

สิ่งที่ควรตรวจสอบ:

- [] RTL-SDR ถูกตรวจพบโดยระบบ
- [] ไม่มี error messages
- [] Ready สำหรับ Lab 2

LAB 2: การรับสัญญาณ DAB+ พื้นฐาน (25 นาที)

การติดตั้ง welle.io

```
# ติดตั้ง welle.io จาก package manager  
sudo apt install welle.io
```

การตั้งค่า welle.io

- ☐ เปิดโปรแกรม welle.io
- ☐ เลือก Input Device: RTL-SDR
- ☐ ตั้งค่า Gain: Auto หรือ 20-30 dB

ความถี่ DAB+ ในประเทศไทย (ตาม NBTC)

- ☐ **Channel 5C:** 178.352 MHz (Bangkok, Pattaya, Hua Hin)
- ☐ **Channel 6C:** 185.360 MHz (National Network)
- ☐ **Channel 7C:** 192.352 MHz (เชียงใหม่, ภาคใต้)
- ☐ **Channel 8C:** 199.360 MHz (Bangkok, Pattaya, Hua Hin)

ขั้นตอนการ Scan

- ☐ เลือก "Band III" (174-230 MHz)
- ☐ กด "Automatic Scan"

LAB 3: Command Line Tools สำหรับ DAB+ (15 นาที)

RTL-SDR Command Line Tools

1. rtl_test - ทดสอบ hardware

```
# ทดสอบ RTL-SDR dongle
rtl_test -t
```

```
# ทดสอบ sample rate
rtl_test -s 2048000
```

- [] ตรวจสอบ PPM error
- [] ดู dropped samples

2. rtl_power - Spectrum Analysis

```
# Scan ย่านความถี่ DAB+ (5 นาที)
rtl_power -f 174M:230M:8k -g 30 -i 10 dab_spectrum.csv
```

3. rtl_fm - FM Demodulation

```
# ฟัง FM radio ปกติ
rtl_fm -M fm -f 101.5M -s 200000 -r 48000 | aplay -r 48000 -f S16_LE
```

4. DAB+ Signal Information

```
# ใช้ rtl_sdr capture raw data
rtl_sdr -f 185360000 -s 2048000 -n 2048000 dab_signal.raw

# ดูข้อมูล file
ls -lh dab_signal.raw
```


LAB 4: ETISnoop - การวิเคราะห์ DAB+ Stream (15 นาที)

การติดตั้ง ETISnoop

1. ติดตั้ง Dependencies

```
sudo apt install build-essential cmake libfftw3-dev librtlsdr-dev
```

2. ดาวน์โหลดและ Compile ETISnoop

```
# Clone repository
git clone https://github.com/JvanKatwijk/eti-snoop
cd eti-snoop

# Build
mkdir build && cd build
cmake ..
make
```

3. การใช้งาน ETISnoop

```
# รัน ETISnoop กับ RTL-SDR
```

สิ่งที่ต้องตรวจสอบใน ETISnoop

1. Ensemble Information

- [] Ensemble Label
- [] Country Code
- [] ECC (Extended Country Code)
- [] Ensemble ID

2. Service Information

- [] Service Labels
- [] Service IDs
- [] Program Types
- [] Bit Rates

LAB 5: การตรวจสอบและวิเคราะห์สัญญาณขั้นสูง (10 นาที)

รายงานผลการทดสอบรวม

ตาราง checklist รวม:

ความถี่ (MHz)	welle.io	Command Line	ETIS
178.352 (5C)			
185.360 (6C)			
192.352 (7C)			
199.360			

ข้อมูลที่ได้จากการทดสอบ

- ☐ Coverage Area: พื้นที่ที่รับสัญญาณได้
- ☐ Service Availability: บริการที่มีในแต่ละ ensemble
- ☐ Technical Quality: คุณภาพทางเทคนิค
- ☐ Comparison Data: เปรียบเทียบระหว่าง tools

Automation Scripts

```
#!/bin/bash
echo "=== DAB+ Automated Test ==="
echo "1. Testing Hardware..."
rtl_test -t

echo "2. Scanning Spectrum..."
rtl_power -f 174M:230M:8k -g 30 -i 5 scan_$(date +%Y%m%d_%H%M).csv

echo "3. Testing Each Channel..."
for freq in 178352000 185360000 192352000 199360000; do
```

การแก้ไขปัญหาและ Troubleshooting

ปัญหาที่พบบ่อยและการแก้ไข

1. RTL-SDR ไม่ทำงาน

- ☐ ตรวจสอบ USB connection
- ☐ ใช้คำสั่ง `lsusb` ดู device
- ☐ รีบูตระบบ
- ☐ ตรวจสอบ driver installation

2. Command Line Tools Error

- ☐ ตรวจสอบ PATH environment
- ☐ ใช้ `which rtl_test` หา location
- ☐ Re-install rtl-sdr package

4. ไม่พบสัญญาณ DAB+

- ☐ ตรวจสอบตำแหน่งเสาอากาศ (วางแนวตั้ง)
- ☐ ลองย้ายไปใกล้หน้าต่าง
- ☐ เปลี่ยน gain setting ใน tools ต่างๆ
- ☐ ทดสอบใน welle.io ก่อน

5. เสียงไม่ดี/กระตุก

- ☐ ตรวจสอบ CPU usage (`top` command)
- ☐ ลด sample rate ถ้าจำเป็น
- ☐ ปิดโปรแกรมอื่นที่ไม่จำเป็น



เคล็ดลับเพิ่มเติม

LAB 6: สร้าง DAB+ Signal Analyzer (Advanced)

Advanced Analysis

- OFDM Structure วิเคราะห์
- SNR, MER, BER คุณภาพสัญญาณ
- Constellation Diagram I/Q แสดงผล
- Waterfall Plot spectrum ตามเวลา

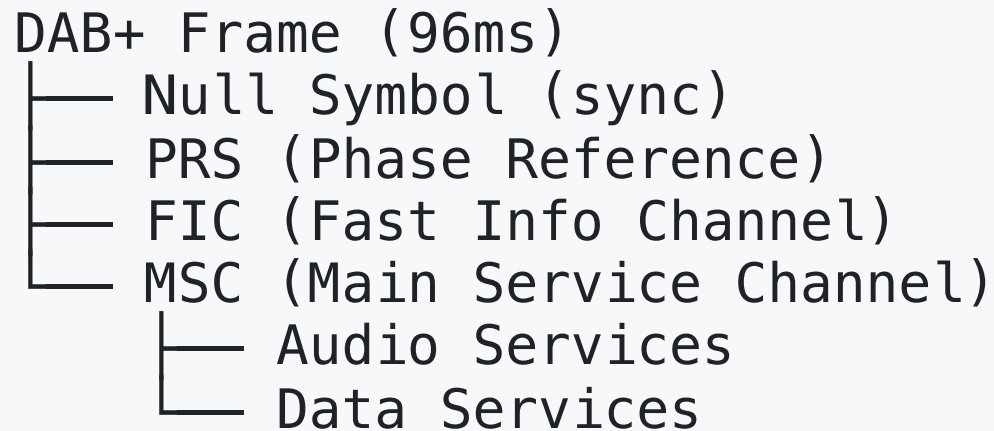
Visualization & Reports

- Real-time Metrics LCD displays
- Professional Reports PDF generation
- Data Export CSV, JSON formats
- Advanced Matplotlib integration

 ผลลัพธ์: Professional DAB+ Signal Analyzer

เทคโนโลยี DAB+ เบื้องลึก

DAB+ Signal Structure



OFDM Technology

- 2048 Carriers ใช้พร้อมกัน
- Guard Interval ป้องกัน multipath
- DQPSK Modulation ทนต่อ noise
- Error Correction Reed-Solomon

 ความเข้าใจ: จากพื้นฐานไปถึงระดับ Professional RF Engineer

การพัฒนาด้วย PyQt5

Touch-Friendly GUI

```
# ปุ่มขนาดใหญ่
button.setMinimumSize(120, 60)

# Font สำหรับหน้าจอ 7"
font = QFont()
font.setPointSize(14)

# CSS Styling
button.setStyleSheet("""
    QPushButton {
        border-radius: 8px;
        background: #3498db;
        color: white;
        font-weight: bold;
    }

```

Signals & Slots

```
# Built-in signals
button.clicked.connect(self.on_click)
slider.valueChanged.connect(self.update_value)

# Custom signals
class MyWidget(QThread):
    data_ready = pyqtSignal(dict)

    def emit_data(self):
        self.data_ready.emit({'value': 42})

```



การประมวลผลสัญญาณ (DSP)

NumPy & SciPy

```
import numpy as np
from scipy import signal

# FFT Analysis
fft_result = np.fft.fft(iq_samples)
frequencies = np.fft.fftfreq(len(samples), 1/sample_rate)

# Power Spectrum
power_db = 20 * np.log10(np.abs(fft_result))

# Signal Quality
snr = signal_power / noise_power
```

Real-time Visualization

```
import matplotlib.pyplot as plt
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg

class SpectrumPlot(FigureCanvasQTAgg):
    def update_spectrum(self, freq, power):
        self.axes.clear()
        self.axes.plot(freq/1e6, power)
        self.draw()
```

การแก้ไขปัญหาทั่วไป

RTL-SDR Issues

```
# ตรวจสอบการเชื่อมต่อ
lsusb | grep RTL

# แก้ driver conflicts
sudo modprobe -r dvb_usb_rtl28xxu
lsmod | grep dvb

# Permissions
sudo usermod -a -G plugdev $USER
```

 เคล็ดลับ: อ่านคู่มือแต่ละ LAB ก่อนเริ่มโค้ด

Audio Issues

```
# ตั้งค่าเสียงออก 3.5mm
sudo raspi-config
# Advanced Options > Audio > Force 3.5mm


# ทดสอบเสียง
speaker-test -t wav -c 2

# PulseAudio restart
pulseaudio -k
```


เส้นทางการเรียนรู้

ระดับเริ่มต้น (Tools-based)


1. **Lab 0:** PyQt5 พื้นฐาน
2. **Lab 1:** RTL-SDR เบื้องต้น (20 นาที)
3. **Lab 2:** DAB+ รับฟัง (25 นาที)
4. **Lab 3:** Command Line Tools (15 นาที)
5. **Lab 4:** ETISnoop Analysis (15 นาที)


 เวลา: ~2 ชั่วโมง

 เป้าหมาย: เรียนรู้ DAB+ tools พื้นฐาน

ระดับสูง (Development-based)

1. **Lab 1:** RTL-SDR Setup + Traps (2 ชม.)
2. **Lab 2:** welle.io Integration + Traps (2 ชม.)
3. **Lab 3:** Spectrum Analysis + Traps (2 ชม.)
4. **Lab 4:** Station Scanner + Traps (2 ชม.)
5. **Lab 5:** Program Recorder + Traps (2 ชม.)
6. **Lab 6:** Signal Analyzer + Traps (2 ชม.)

 เวลา: ~12 ชั่วโมง

 เป้าหมาย: Professional DAB+ Applications

ผลลัพธ์ที่ได้รับ

แอปพลิเคชันที่สร้างได้

เริ่มต้น (Tools-based):

- Basic DAB+ Reception
- Command Line Analysis

ขั้นสูง (Development-based):

- RTL-SDR Hardware Controller
- DAB+ Station Scanner
- Program Recorder with Scheduling
- Real-time Spectrum Analyzer
- Professional Signal Analyzer

ความรู้ที่ได้รับ

เริ่มต้น:

- DAB+ Technology พื้นฐาน
- Command Line Tools

ขั้นสูง:

- Python & PyQt5 GUI Development
- RF & DSP Signal Processing
- Real-time Audio Processing
- Database & Threading
- OFDM & Machine Learning
- Professional RF Analysis

แหล่งข้อมูลเพิ่มเติม

Documentation

- [welle.io GitHub](#)
- [RTL-SDR.com](#)
- [PyQt5 Docs](#)
- [DAB+ Standard \(ETSI\)](#)

Learning Resources

- GNU Radio สำหรับ SDR ขั้นสูง
- DSP Course Signal Processing
- RF Engineering คลื่นวิทยุ
- Embedded Linux สำหรับ IoT

Next Steps - ขั้นตอนต่อไป

พัฒนาเพิ่มเติม

- **Web Interface** ควบคุมผ่าน browser
- **Mobile App** Android/iOS remote
- **Cloud Integration** upload recordings
- **AI/ML** automatic classification

 จาก Hobby Project → Professional Career

Career Paths

- **RF Engineer** วิศวกรคลื่นวิทยุ
- **SDR Developer** Software Defined Radio
- **IoT Developer** Internet of Things
- **Embedded Systems** ระบบฝังตัว

ขอขอบคุณและสนับสนุน

 ขอขอบคุณที่เรียนรู้กับเรา!

DAB+ Labs เป็นโครงการ Open Source
สำหรับการศึกษาและพัฒนาเทคโนโลยี

 ขอให้สนุกกับการเรียนรู้!

ติดต่อและสนับสนุน

-  **Issues:** GitHub Issues
-  **Email:** project contact
-  **Community:** Forum discussion
-  **Star:** ถ้าชอบโครงการ

MIT License - ใช้ได้อย่างอิสระ

สรุป: การเดินทาง DAB+ Learning

สิ่งที่เราได้เรียนรู้:

- DAB+ Technology จาก 0 ถึง Hero
- Python & PyQt5 สำหรับ Professional GUI
- RTL-SDR & RF Engineering ด้วยมือ
- Project Development จาก Idea ถึง Working App

Achievement Unlocked:

 DAB+ Expert |  Python GUI Master |  RF Engineer |  Maker

 พร้อมสำหรับการผจญภัยใหม่แล้ว! 