

Smart Alarming IoT-Based System Using Raspberry Pi 4

Message Transmission

Name: Siddharth Karmokar

Roll No: 123CS0061

1. Introduction

This project focuses on **message transmission** using **Blynk**, **Twilio**, and an **SPI TFT display** integrated with a **Raspberry Pi 4**. The system allows a user to send a message via the Blynk app, which is received by the Raspberry Pi, displayed on the **1.8" SPI TFT LCD**, and forwarded as an SMS using **Twilio API**. Challenges included setting up SPI communication, ensuring reliable message transmission, and handling delays in network communication. The implementation also required GPIO handling for LED notification upon message reception.

2. Components Used

- **Raspberry Pi 4**
 - **1.8" SPI TFT LCD (ST7735)**
 - **Internet Connectivity (WiFi)**
 - **Twilio Account for SMS transmission**
 - **Blynk IoT Platform**
 - **LED Indicator**
 - **Power Supply (5V, 2.5A)**
-

3. System Architecture

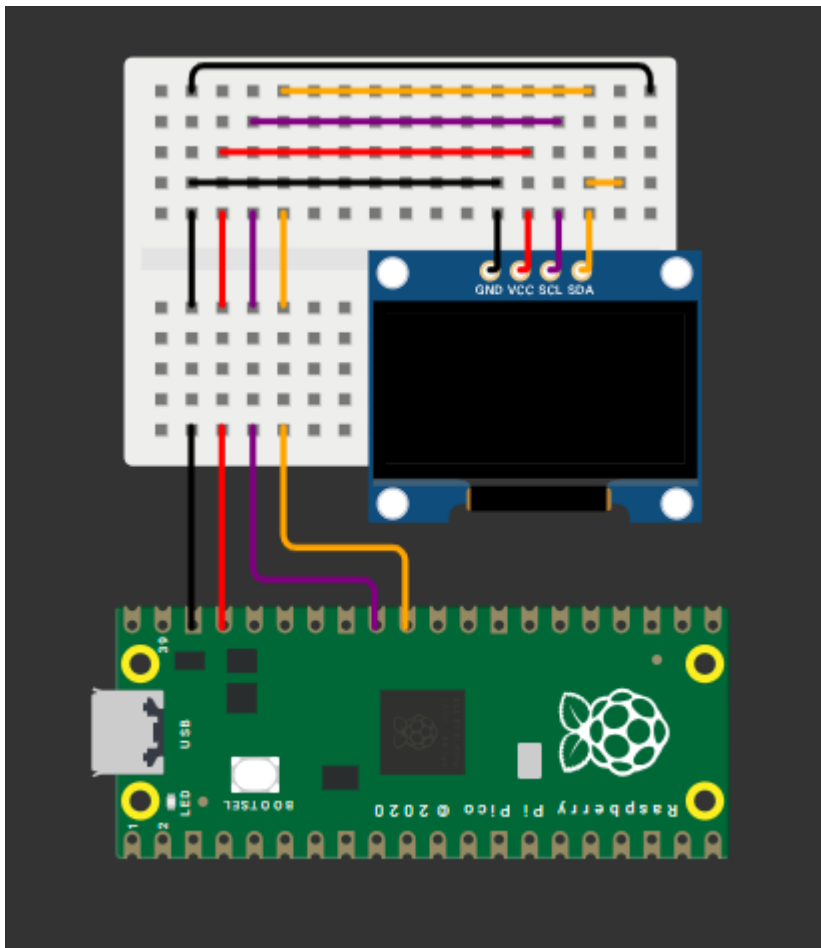
- **User inputs message** via the **Blynk mobile app**.
 - **Raspberry Pi receives** the message and **displays it** on the **TFT screen**.
 - The same message is **forwarded via Twilio** as an SMS to a designated phone number.
 - An **LED blinks** to indicate message reception.
 - The system ensures **real-time message transmission** using the **SPI protocol** for display and **Twilio API** for SMS.
-

4. Software and Tools Required

- **Raspberry Pi OS (Debian-based)**
- **Python 3.9**
- **Blynk Python Library**
- **Twilio API**
- **Pillow (PIL) for Display Handling**

- ST7735 Python Library for SPI TFT
 - RPi.GPIO for GPIO Handling
-

5. Circuit Diagram



6. Code

```
import blynklib
from twilio.rest import Client
from PIL import Image, ImageDraw, ImageFont
import ST7735 # Updated to ST7735 for 1.8" SPI TFT
import time
import RPi.GPIO as GPIO

# Blynk Auth Token
BLYNK_AUTH = "YOUR_BLYNK_AUTH_TOKEN"
blynk = blynklib.Blynk(BLYNK_AUTH)

# Twilio Credentials
ACCOUNT_SID = "YOUR_TWILIO_SID"
AUTH_TOKEN = "YOUR_TWILIO_AUTH_TOKEN"
TO_NUMBER = "+1234567890"
FROM_NUMBER = "+0987654321"

# LED Pin Setup
LED_PIN = 17 # Change to your actual LED GPIO pin
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)

def send_sms(message):
    """Send message using Twilio"""
    client = Client(ACCOUNT_SID, AUTH_TOKEN)
    client.messages.create(to=TO_NUMBER, from_=FROM_NUMBER, body=message)
    print("SMS sent!")

def display_on_tft(message):
    """Display message on SPI TFT LCD (ST7735)"""
    disp = ST7735.ST7735(
        port=0, cs=0, dc=25, backlight=18, rotation=180
    )
    disp.begin()

    img = Image.new("RGB", (128, 160), color=(0, 0, 0)) # Adjusted for 1.8"
display resolution
    draw = ImageDraw.Draw(img)
    font = ImageFont.load_default()

    draw.text((10, 70), message, font=font, fill=(255, 255, 255))
    disp.display(img)
    print("Message displayed on TFT!")

@blynk.on("V1") # Virtual pin V1 for receiving messages
def read_message(value):
    message = value[0]
    print(f"Received from Blynk: {message}")
```

```
# Blink LED to indicate message received
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(1)
GPIO.output(LED_PIN, GPIO.LOW)

display_on_tft(message)
send_sms(message)

print("Waiting for messages from Blynk...")
while True:
    blynk.run()
    time.sleep(0.1)
```