สรุปองค์ความรู้

(เล่มที่ 2)

การจัดการความรู้ (KM) ประจำปี พ.ศ. 2568

ศูนย์อุตุนิยมวิทยาภาคตะวันออกเฉียงเหนือตอนล่าง

เรื่อง

คู่มือการใช้งานโปรแกรมจัดทำภาพอินโฟกราฟิก เรดาร์ตรวจอากาศ เพื่อการแจ้งเตือนในพื้นที่ (โปรแกรม ClickRain)

โดย คณะทำงานการจัดการความรู้ ปีงบประมาณ พ.ศ.2568

ศูนย์อุตุนิยมวิทยาภาคตะวันออกเฉียงเหนือตอนล่าง

สารบัญ

1. บทนำ 2

2. ภาพรวมของโปรแกรม 3

3. ขั้นตอนการติดตั้งและเตรียมข้อมูล 4

4. ส่วนประกอบของโปรแกรม 5

5. วิธีใช้งานโปรแกรม 7

6. การนำความรู้ไปใช้จริง (KM: Knowledge Application) 12

7. สรุป 12

8. ข้อจำกัดและข้อเสนอแนะเพิ่มเติม 13

9. ขั้นตอนการสร้างโปรแกรม ClickRain 13

10. ภาคผนวก 18

ตารางสรุปการเชื่อมโยง: ปุ่ม ⟶ Module ⟶ ฟังก์ชัน 19

1.ปุ่ม ActiveX 21

2. Module 35

3. User Defined Function (UDF) 60

1. บทนำ

1.1 ที่มาและความสำคัญของโปรแกรม

ในกระบวนการรายงานและติดตามสภาวะอากาศ การวิเคราะห์กลุ่มฝนจากภาพเรดาร์ตรวจอากาศและจัดทำให้อยู่ในรูปแบบอินโฟกราฟิก ถือเป็นองค์ประกอบสำคัญ เนื่องจากสามารถสื่อสารข้อมูลไปยังประชาชนและผู้เกี่ยวข้องได้อย่างรวดเร็ว และเข้าใจง่าย อย่างไรก็ตาม กระบวนการจัดทำภาพวิเคราะห์กลุ่มฝนจากเรดาร์ในปัจจุบันยังประกอบด้วยหลายขั้นตอน โดยระยะเวลาในการจัดทำจะขึ้นอยู่กับประสบการณ์ของผู้ปฏิบัติงาน

จากแนวคิดในการลดภาระงานที่ซ้ำซ้อน เพิ่มความแม่นยำ และสร้างมาตรฐานในการจัดทำอินโฟกราฟิกวิเคราะห์กลุ่มฝนจากเรดาร์ตรวจอากาศ จึงได้มีการพัฒนา **“โปรแกรมวิเคราะห์ฝนจากภาพเรดาร์แบบคลิก (ClickRain)”** ขึ้น เพื่อใช้เป็นเครื่องมือช่วยให้เจ้าหน้าที่สามารถวิเคราะห์พื้นที่ฝน ความรุนแรง และแนวโน้มการเคลื่อนตัวของกลุ่มฝน พร้อมจัดทำภาพได้อย่างสะดวก รวดเร็ว และเป็นระบบ

โปรแกรมนี้ได้รับการออกแบบให้สามารถดึงภาพเรดาร์จากเว็บไซต์ของกรมอุตุนิยมวิทยา และแสดงขอบเขตพื้นที่อำเภอบนภาพเรดาร์ได้โดยตรง เมื่อผู้ใช้คลิกที่กลุ่มฝนในบริเวณอำเภอ โปรแกรมจะแสดงชื่ออำเภอและจัดกลุ่มตามจังหวัดโดยอัตโนมัติ ผู้ใช้สามารถเลือกปรับระดับความรุนแรงของกลุ่มฝน ใส่ลูกศรแสดงทิศทางการเคลื่อนตัว และสร้างอินโฟกราฟิกวิเคราะห์กลุ่มฝนที่สามารถนำไปเผยแพร่ต่อผ่านการวาง (paste) บนโซเชียลมีเดียต่าง ๆ ได้ทันที

1.2 วัตถุประสงค์ของคู่มือ

คู่มือนี้จัดทำขึ้นเพื่อ

- เผยแพร่ความรู้เกี่ยวกับการพัฒนาและใช้งานโปรแกรม ClickRain
- เป็นเอกสารอ้างอิงสำหรับนักพัฒนาหรือผู้ที่ต้องการนำโปรแกรมไปต่อยอด
- ใช้เป็นคู่มือสำหรับเจ้าหน้าที่ผู้ปฏิบัติงาน เพื่อให้สามารถใช้งานโปรแกรมได้อย่างถูกต้องและมีประสิทธิภาพ

1.3 กลุ่มเป้าหมายผู้ใช้งาน

กลุ่มผู้ใช้งานหลักของโปรแกรม ClickRain ได้แก่

- เจ้าหน้าที่ผู้ปฏิบัติงานในส่วนติดตามสภาวะอากาศ
- โดยเฉพาะศูนย์อุตุนิยมวิทยาภาคตะวันออกเฉียงเหนือตอนล่าง

ผู้ใช้งานไม่จำเป็นต้องมีพื้นฐานด้าน Excel หรือการเขียน VBA แต่อย่างใด เพียงมีทักษะการใช้งานคอมพิวเตอร์พื้นฐานก็สามารถใช้งานโปรแกรมได้ทันที

2. ภาพรวมของโปรแกรม

2.1 แนวคิดเบื้องหลังโปรแกรม

โปรแกรม ClickRain ถูกพัฒนาขึ้นจากแนวคิดที่ต้องการให้ผู้ปฏิบัติงานสามารถจัดทำภาพวิเคราะห์กลุ่มฝนจากภาพเรดาร์ตรวจอากาศได้อย่างสะดวก รวดเร็ว และมีความถูกต้องมากยิ่งขึ้น โปรแกรมนี้ยังสะท้อนเจตนารมณ์ของผู้พัฒนาในการนำความรู้ด้านเทคโนโลยีมาใช้ให้เกิดประโยชน์ต่อเจ้าหน้าที่และประชาชน

2.2 องค์ความรู้เดิมที่นำมาประยุกต์

แนวคิดการคลิกบนภาพเรดาร์แล้วแสดงชื่ออำเภอได้รับแรงบันดาลใจมาจากผลงานของนายปริวัชร นามด้วง นักอุตุนิยมวิทยาชำนาญการ ซึ่งเคยเผยแพร่ไว้ที่เว็บไซต์ของศูนย์อุตุนิยมวิทยาภาคตะวันออกเฉียงเหนือตอนล่าง (https://ubonmet.tmd.go.th/files/tv/KM68/main.html) โดยผู้พัฒนาได้นำแนวคิดดังกล่าวมาปรับปรุงให้เหมาะสมกับการใช้งานจริงใน Excel และเพิ่มประสิทธิภาพด้านความเร็วและความแม่นยำในการระบุพื้นที่ฝน

2.3 จุดเด่นและความแตกต่างของโปรแกรมนี้

- ลดเวลาในการจัดทำภาพอินโฟกราฟิกวิเคราะห์กลุ่มฝน
- ได้ภาพที่มีความเป็นมาตรฐานและเข้าใจง่าย
- ลดข้อผิดพลาดจากการประเมินตำแหน่งอำเภอด้วยตนเอง
- ใช้งานง่าย ไม่จำเป็นต้องมีความรู้พื้นฐาน Excel มาก

2.4 ขอบเขตการใช้งานของโปรแกรม

โปรแกรม ClickRain ออกแบบมาเพื่อใช้งานร่วมกับภาพเรดาร์ตรวจอากาศจากสถานีเรดาร์สกลนครของกรมอุตุนิยมวิทยา โดยใช้ภาพจากลิงค์ https://weather.tmd.go.th/skn/skn240_latest.jpg ซึ่งครอบคลุมพื้นที่ภายในรัศมี 240 กิโลเมตร จากสถานี

โปรแกรมสามารถใช้วิเคราะห์กลุ่มฝนจากภาพเรดาร์ตรวจอากาศที่เผยแพร่ทุก ๆ 15 นาที และสามารถเรียกดูภาพเคลื่อนไหว (Radar Loop) เพื่อประเมินทิศทางของกลุ่มฝนได้ นอกจากนี้ยังสามารถแทรกลูกศรเพื่อแสดงทิศทางของกลุ่มฝนได้อย่างสะดวกตามการประเมินของผู้ปฏิบัติงาน

3. ขั้นตอนการติดตั้งและเตรียมข้อมูล

3.1 ความต้องการของระบบ

โปรแกรม ClickRain แนะนำให้ใช้งานบน Microsoft Excel เวอร์ชัน 2021 ขึ้นไป เพื่อให้สามารถใช้งานฟังก์ชัน VBA และ ActiveX ได้อย่างสมบูรณ์ ทั้งนี้ผู้ใช้งานต้องเปิดการใช้งาน Macro และอนุญาต ActiveX Control ก่อนใช้งาน

3.2 การเปิดใช้งาน Macro และ ActiveX

- เมื่อเปิดไฟล์โปรแกรมครั้งแรก ให้กด "Enable Content" เพื่อเปิดใช้งาน Macro
- ระบบใช้ปุ่มควบคุมแบบ ActiveX ผู้ใช้จำเป็นต้องอนุญาตให้ Excel รัน ActiveX Control

3.3 องค์ประกอบในไฟล์โปรแกรม

- โปรแกรมมีการฝัง shape เขตอำเภอไว้เรียบร้อยแล้ว ไม่ต้องติดตั้งหรือเพิ่มไฟล์ภายนอกเพิ่มเติม
- โปรแกรมจะโหลดภาพเรดาร์ตรวจอากาศจากลิงค์ของกรมอุตุนิยมวิทยาโดยอัตโนมัติ ไม่จำเป็นต้องดาวน์โหลดหรือจัดวางภาพเอง
- โปรแกรมไม่มี template แยก ผู้ใช้สามารถเปิดไฟล์หลักและใช้งานได้ทันที

3.4 การเตรียมข้อมูลเบื้องต้น

ผู้ใช้งานไม่จำเป็นต้องตั้งค่าใด ๆ ก่อนใช้งาน เพียงเปิดไฟล์แล้วคลิกปุ่มตามลำดับในคู่มือการใช้งาน (รายละเอียดในหัวข้อถัดไป) โปรแกรมจะทำงานอัตโนมัติตามลำดับขั้นที่ออกแบบไว้

4. ส่วนประกอบของโปรแกรม

4.1 โครงสร้างของชีตและพื้นที่การทำงาน

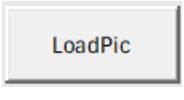
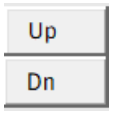
โปรแกรม ClickRain ทำงานบน Excel โดยใช้เพียง 1 ชีตหลักที่ประกอบด้วยพื้นที่ทำงาน 2 ส่วนสำคัญ ได้แก่:

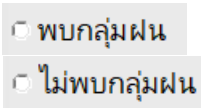
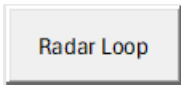
- พื้นที่แสดงภาพเรดาร์และผลวิเคราะห์: อยู่ด้านซ้ายของชีต ใช้สำหรับแสดงภาพเรดาร์พร้อม shape เขตอำเภอ และผลวิเคราะห์ที่ถูกจัดวางเพื่อจัดทำเป็นภาพอินโฟกราฟิกสำหรับเผยแพร่
- แถบควบคุมการทำงาน (Control Panel): อยู่ด้านขวาของชีต ประกอบด้วยปุ่มควบคุมลำดับการใช้งาน และตัวเลือกต่าง ๆ ที่ออกแบบไว้ในรูปแบบ ActiveX Control

4.2 ส่วนประกอบหลักของหน้าจอ

องค์ประกอบ	รายละเอียด
ภาพเรดาร์	แสดงอยู่ตรงกลาง/ด้านซ้ายของชีต เป็นภาพจากลิงค์เรดาร์ของสถานีสกลนคร ขนาดภาพถูกดึงโดยอัตโนมัติ
Shape เขตอำเภอ	ฝังอยู่ในชีตโดยตรง ประกอบด้วย shape หลายชิ้นแทนอำเภอต่าง ๆ ในรัศมี 240 กม.
ลูกศรแสดงทิศทางกลุ่มฝน	ผู้ใช้สามารถแทรกลงในภาพได้เพื่อแสดงทิศทางการเคลื่อนที่ของกลุ่มฝน
แถบสีระดับฝน	แสดงอยู่ด้านข้างซ้ายของภาพเรดาร์ เป็น Legend แสดงระดับฝนตามค่าสีจากเรดาร์

4.3 ปุ่มและคำสั่งต่าง ๆ

หมายเลข	ชื่อปุ่ม / องค์ประกอบ	หน้าที่
1	LoadPic 	ดึงภาพเรดาร์จากลิงค์เข้าสู่ชีตอัตโนมัติ
2	Up / Dn 	ปรับเวลาแสดงภาพให้ตรงกับภาพเรดาร์ที่ต้องการใช้งาน

3	พบกลุ่มฝน / ไม่พบกลุ่มฝน 	ระบุสถานะการตรวจพบกลุ่มฝนในรอบการรายงานนั้น
3.1–3.3	ระดับฝน (ฝนอ่อน–หนัก) 	เลือกอำเภอที่พบฝนตามระดับความแรง
4	Radar Loop (Toggle) 	แสดงหรือซ่อนภาพเรดาร์ย้อนหลังแบบต่อเนื่อง เพื่อดูการเคลื่อนที่ของกลุ่มฝน
5	Direction (Toggle) 	เปิดหรือปิดโหมดการวาดลูกศร
6	Build 	สร้างภาพวิเคราะห์ฝน และบันทึกภาพอัตโนมัติไปยังโฟลเดอร์ที่กำหนด พร้อมทั้งคัดลอกภาพไว้ในคลิปบอร์ดเพื่อให้สามารถวาง (Ctrl + V) ได้ทันที
-	Path 	แสดงพาธของโฟลเดอร์ที่ใช้จัดเก็บภาพที่สร้าง

4.4 ActiveX Control ที่ใช้

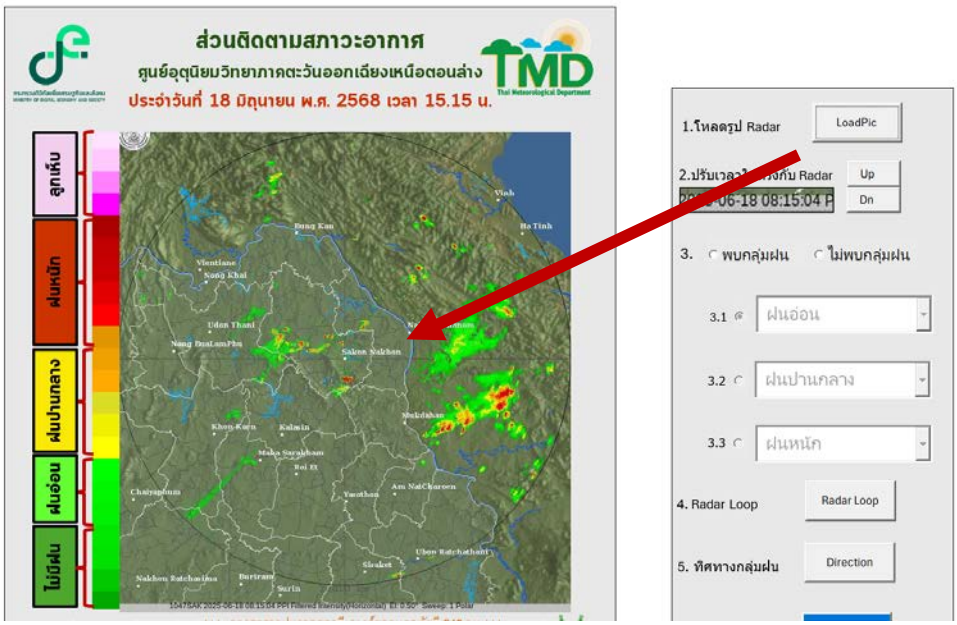
- CommandButton (ปุ่มทั้งหมด เช่น LoadPic, Up, Dn, Build)
- OptionButton (พบกลุ่มฝน / ไม่พบกลุ่มฝน)
- ComboBox (สำหรับเลือกอำเภอในแต่ละระดับฝน)
- ToggleButton (Radar Loop และ Direction)

5. วิธีใช้งานโปรแกรม

โปรแกรม ClickRain ได้ออกแบบส่วนควบคุมให้อยู่ภายในแถบ Control Panel ทางด้านขวาของชีต Excel โดยผู้ใช้งานสามารถปฏิบัติตามลำดับขั้นตอนต่อไปนี้เพื่อสร้างภาพวิเคราะห์ฝนจากเรดาร์อย่างสมบูรณ์

ขั้นตอนที่ 1: โหลดรูปเรดาร์

ปุ่ม: “LoadPic”

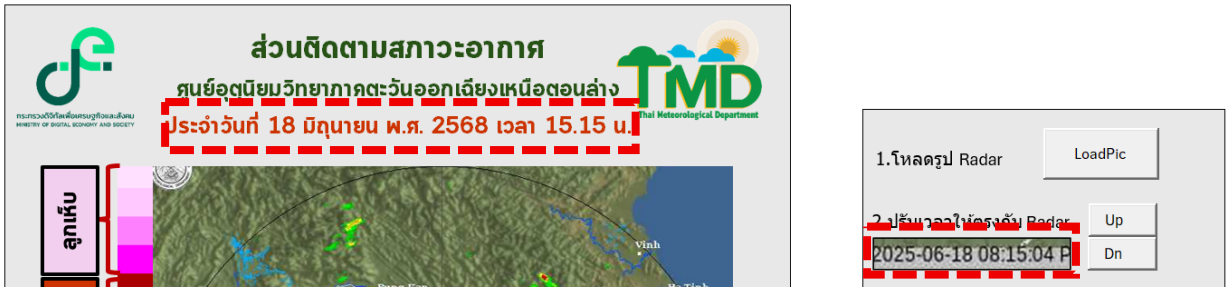


ภาพที่ 1 แสดงผลลัพธ์จากการกดปุ่ม LoadPic ซึ่งโหลดภาพเรดาร์จากเว็บไซต์และปรับขนาดให้พอดีกับพื้นที่แสดงผลโดยอัตโนมัติ

- โปรแกรมจะดึงภาพเรดาร์ล่าสุดจากลิงค์ของกรมอุตุนิยมวิทยา (สถานีเรดาร์สกลนคร) มาแสดงโดยอัตโนมัติ
- ผู้ใช้งานไม่จำเป็นต้องดาวน์โหลดหรือวางภาพเอง

ขั้นตอนที่ 2: ปรับเวลาให้ตรงกับภาพเรดาร์

ปุ่ม: “Up”, “Dn”

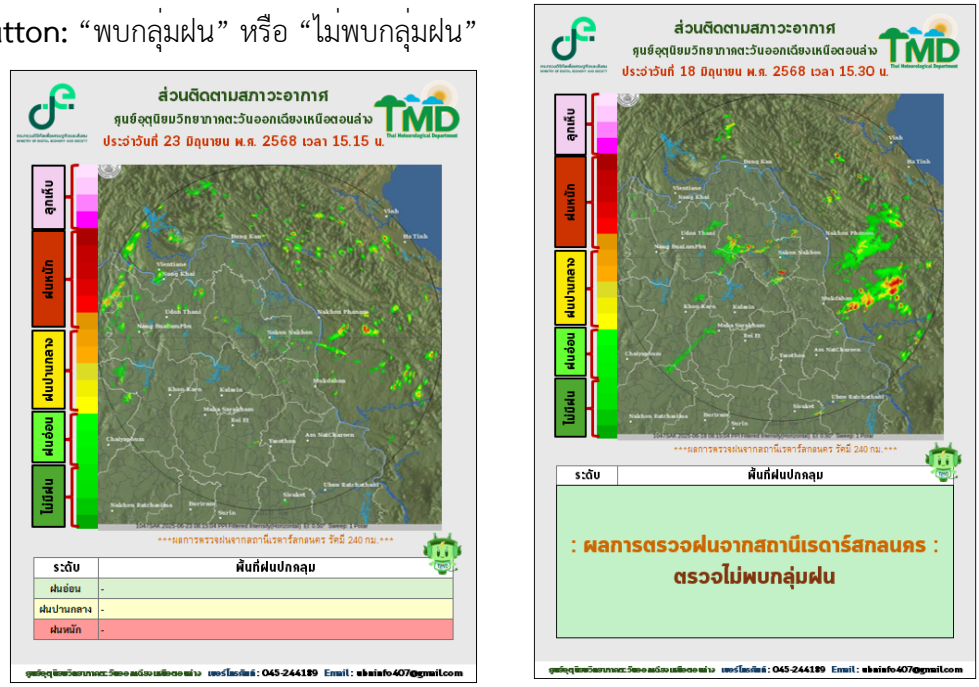


ภาพที่ 2 แสดงภาพการปรับเวลาให้สอดคล้องกับภาพเรดาร์

- ใช้ปรับเวลาที่จะแสดงบนภาพ เพื่อให้ตรงกับเวลาของภาพเรดาร์ที่โหลดมา
- ปุ่ม “Up” ใช้เลื่อนเวลาให้เร็วขึ้น
- ปุ่ม “Dn” ใช้เลื่อนเวลาให้ย้อนกลับ

ขั้นตอนที่ 3: ระบุการพบกลุ่มฝน

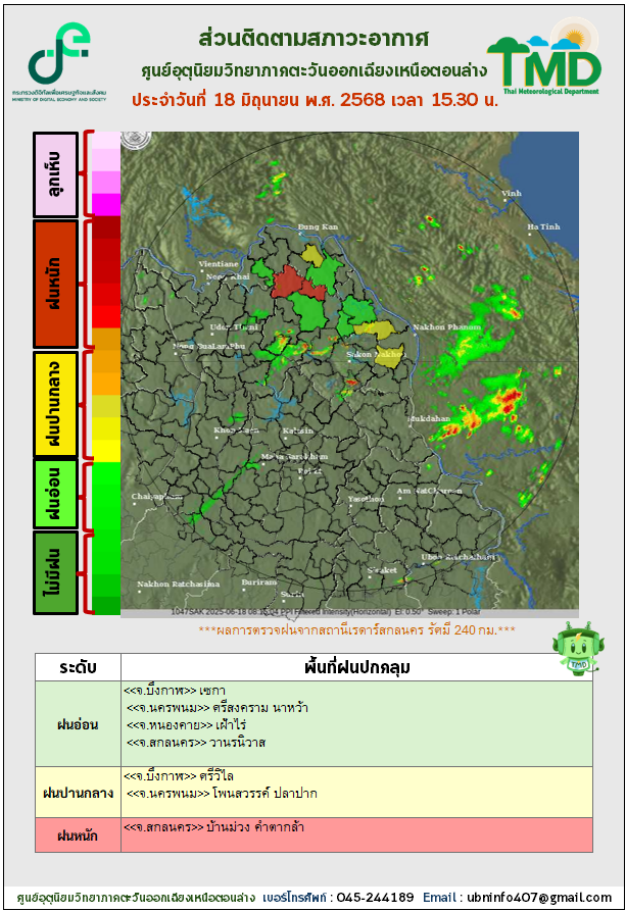
OptionButton: “พบกลุ่มฝน” หรือ “ไม่พบกลุ่มฝน”



ภาพที่ 3 แสดงเทมเพลตที่แตกต่างกันระหว่างเลือกปุ่ม“พบกลุ่มฝน” หรือ “ไม่พบกลุ่มฝน”

- หากพบกลุ่มฝน ให้เลือก “พบกลุ่มฝน” แล้วกรอกข้อมูลอำเภอที่มีฝนในขั้นถัดไป
- หากไม่พบฝน ให้เลือก “ไม่พบกลุ่มฝน” โปรแกรมจะไม่ต้องใส่รายละเอียดระดับฝน

ขั้นตอนที่ 4: ระบุอำเภอที่มีฝน (ถ้ามี)



ภาพที่ 4 แสดงการคลิกเลือกฝนตามระดับความรุนแรงบนวัตถุรูปร่าง (Shape)

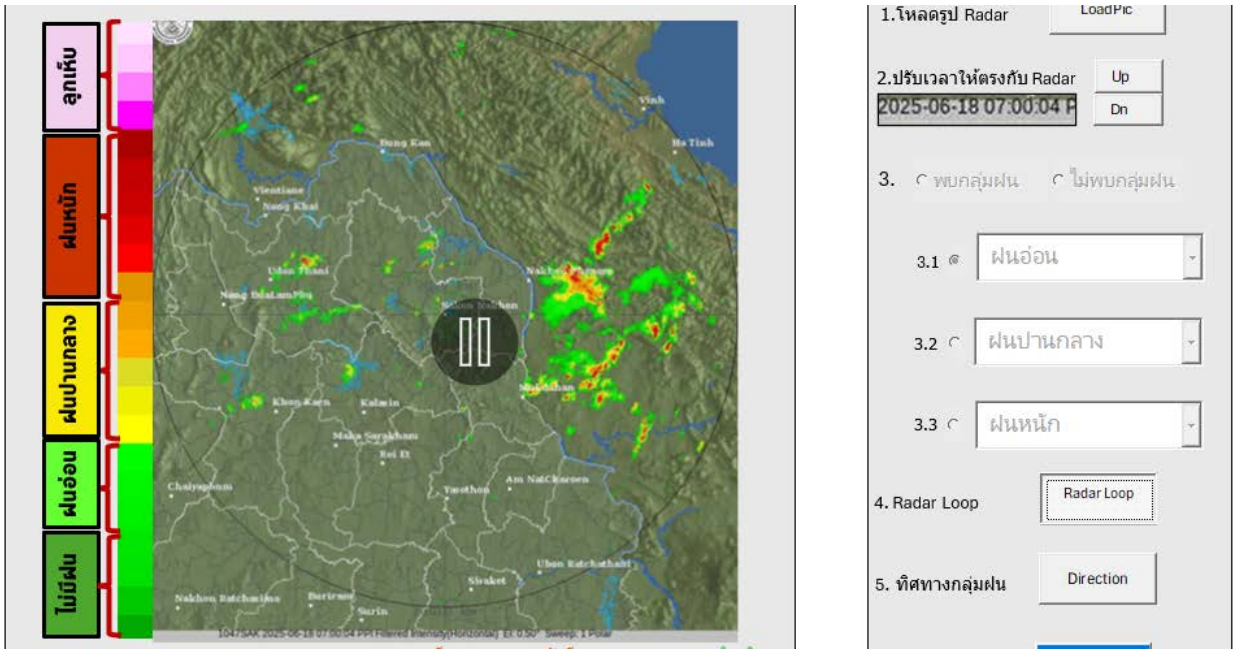
กล่องเลือกข้อมูล (ComboBox) :

- 3.1 เลือกอำเภอที่มี **ฝนอ่อน**
- 3.2 เลือกอำเภอที่มี **ฝนปานกลาง**
- 3.3 เลือกอำเภอที่มี **ฝนหนัก**

เมื่อคลิกที่ shape ของอำเภอ โปรแกรมจะเปลี่ยนสี shape ตามระดับความรุนแรงของฝนโดยอัตโนมัติ เพื่อความสะดวกในการจดจำ และหากต้องการลบชื่ออำเภอ สามารถคลิกซ้ำที่อำเภอเดิมเพื่อยกเลิกการเลือกได้

ขั้นตอนที่ 5: แสดง Radar Loop (ถ้าต้องการ)

ToggleButton: “Radar Loop”

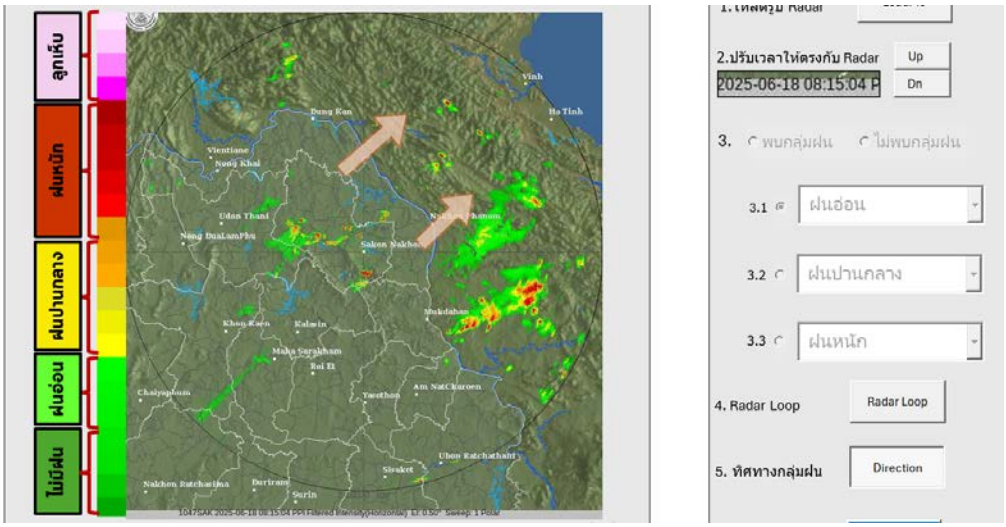


ภาพที่ 5 แสดงการคลิกเปิดภาพเรดาร์เคลื่อนไหว (Radar Loop)

- ใช้เปิด/ปิดการแสดงภาพเรดาร์ย้อนหลัง (ย้อนหลังหลายช่วงเวลา)
- ก่อนแสดงภาพเรดาร์ย้อนหลัง โปรแกรมจะปรับช่องของข้อมูลอำเภอให้พอดีกับข้อความโดยอัตโนมัติ

ขั้นตอนที่ 6: ใส่ลูกศรแสดงทิศทางกลุ่มฝน

ToggleButton: “Direction”

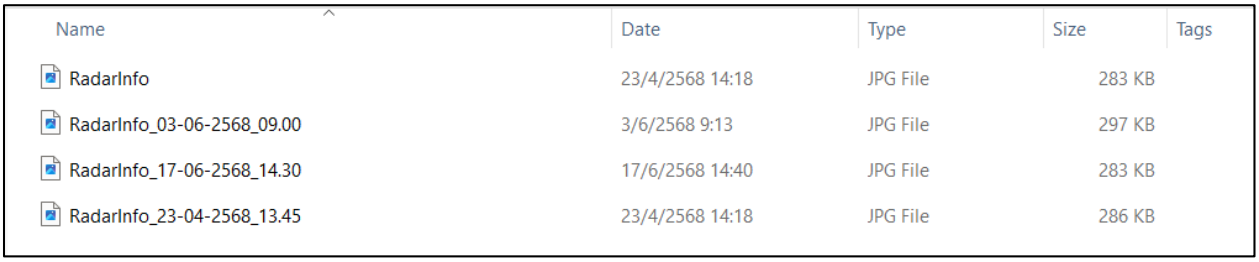


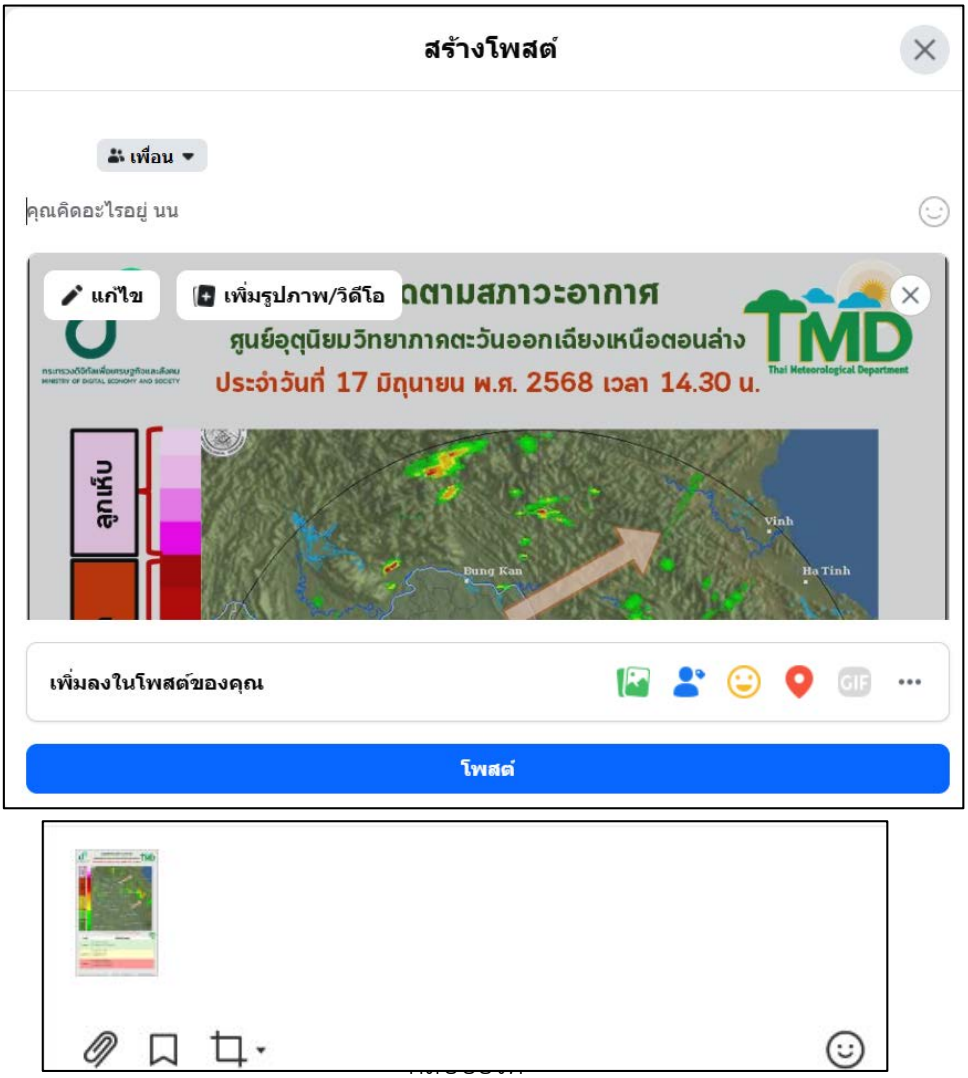
ภาพที่ 6 แสดงการคลิก 2 ครั้งเพื่อวาดลูกศรแสดงทิศการเคลื่อนที่กลุ่มฝน

- คลิกที่ภาพเรดาร์
- คลิกแรก: เป็นจุดเริ่มต้นของลูกศร
- คลิกที่สอง: เป็นปลายลูกศร
- หากต้องการลบลูกศร ให้คลิกที่ตัวลูกศรโดยตรง

ขั้นตอนที่ 7: สร้างภาพและบันทึก

ปุ่ม: “Build”





ภาพที่ 7 แสดงผลลัพธ์หลังคลิกปุ่ม Build สามารถบันทึกไฟล์และและคัดลอกภาพไปยังคลิปบอร์ด

- โปรแกรมจะจัดองค์ประกอบทั้งหมด (ภาพเรดาร์, อำเภอ, ลูกศร, รายละเอียด) ลงในภาพ
- ภาพที่ได้จะถูก:
    - **บันทึกอัตโนมัติ** ไปยังโฟลเดอร์ที่กำหนด โดยตั้งชื่อไฟล์ตามวันเวลาเรดาร์
    - **คัดลอกลงคลิปบอร์ด** ให้สามารถกด Ctrl + V เพื่อนำไปวางในโปรแกรมอื่นได้ทันที

หมายเหตุ: ควรปฏิบัติตามลำดับปุ่มเพื่อให้โปรแกรมทำงานอย่างถูกต้อง

6. การนำความรู้ไปใช้จริง (KM: Knowledge Application)

6.1 ข้อเสนอแนะสำหรับผู้ใช้งาน

- ควรใช้งานโปรแกรมโดยปฏิบัติตามลำดับของปุ่มใน Control Panel อย่างเคร่งครัด เพื่อให้การทำงานของโปรแกรมสมบูรณ์
- หมั่นตรวจสอบความถูกต้องของเวลาและภาพเรดาร์ก่อนเริ่มวิเคราะห์ทุกครั้ง
- เมื่อพบว่าการคลิกเลือกอำเภอหรือวาดลูกศรผิดพลาด สามารถกดซ้ำที่อำเภอหรือลบลูกศรได้ทันที โดยไม่ต้องเริ่มต้นใหม่ทั้งหมด
- ควรทดลองใช้งานหลายครั้งก่อนนำไปใช้จริง เพื่อความมั่นใจในการใช้งานในสภาวะเวลาจำกัด

6.2 แนวทางการพัฒนาต่อ

- เพิ่มความสามารถในการเลือกสถานีเรดาร์อื่น ๆ นอกเหนือจากสกลนคร
- ออกแบบอินเทอร์เฟซให้สามารถปรับขนาดหรือเลื่อนตำแหน่งภาพเรดาร์ได้แบบยืดหยุ่นมากขึ้น
- อาจพิจารณาเพิ่มระบบการ Export ข้อมูลลงรายงานหรือระบบรายงานอัตโนมัติในอนาคต
- พัฒนาเวอร์ชันที่สามารถใช้งานบนเว็บไซต์ เพื่อให้สะดวกต่อการเข้าถึง และลดอุปสรรคจากการติดตั้งโปรแกรมบนเครื่องผู้ใช้งาน

หากมีข้อเสนอแนะเพิ่มเติมจากผู้ใช้งานในศูนย์ฯ หรือหน่วยงานอื่น สามารถรวบรวมและวิเคราะห์เพิ่มเติมเพื่อเป็นแนวทางในการพัฒนาในระยะยาวต่อไป

7. สรุป

โปรแกรม ClickRain เป็นเครื่องมือที่ถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกแก่เจ้าหน้าที่ผู้ปฏิบัติงานในการจัดทำภาพอินโฟกราฟิกวิเคราะห์กลุ่มฝนจากภาพเรดาร์ โดยเน้นความรวดเร็ว ความถูกต้อง และมาตรฐานในการสื่อสารข้อมูล ใช้งานได้ง่ายแม้ไม่มีความรู้ด้าน Excel หรือ VBA มาก่อน

จากการนำความรู้มาประยุกต์ใช้งานจริงในศูนย์อุตุนิยมวิทยาภาคตะวันออกเฉียงเหนือตอนล่าง พบว่าโปรแกรมช่วยลดเวลาในการทำงาน เพิ่มความแม่นยำในการวิเคราะห์ และสามารถต่อยอดได้อีกหลากหลายรูปแบบในอนาคต

การจัดทำคู่มือฉบับนี้มีเป้าหมายเพื่อเผยแพร่ความรู้ให้บุคลากรสามารถใช้งานโปรแกรมได้อย่างถูกต้อง และเพื่อให้ผู้พัฒนาท่านอื่นสามารถนำไปปรับปรุงหรือประยุกต์ใช้กับงานในลักษณะเดียวกันได้อย่างมีประสิทธิภาพ

8. ข้อจำกัดและข้อเสนอแนะเพิ่มเติม

แม้โปรแกรม ClickRain จะถูกออกแบบให้ใช้งานง่ายและตอบโจทย์การวิเคราะห์กลุ่มฝนจากภาพเรดาร์ แต่ก็ยังมีข้อจำกัดบางประการที่ควรพิจารณาในการพัฒนาเพิ่มเติมในอนาคต ได้แก่:

- โปรแกรมยังรองรับเฉพาะเรดาร์จากสถานีสกลนคร ไม่สามารถเลือกสถานีอื่นได้โดยตรง
- จำเป็นต้องเปิดใช้งาน Excel ที่รองรับ VBA และเปิด Macro/ActiveX ซึ่งอาจมีข้อจำกัดในบางหน่วยงาน
- ไม่มีระบบการตรวจสอบสิทธิ์ผู้ใช้งาน หรือการจัดเก็บประวัติการวิเคราะห์เพื่อใช้ในการตรวจสอบย้อนหลัง

ข้อเสนอแนะเพิ่มเติม

- พัฒนาเวอร์ชันบนเว็บหรือ Web App เพื่อใช้งานได้บนหลากหลายแพลตฟอร์ม
- เชื่อมโยงข้อมูลฝนที่วิเคราะห์แล้วเข้ากับระบบเตือนภัย หรือฐานข้อมูลเชิงสถิติ

9. ขั้นตอนการสร้างโปรแกรม ClickRain

การพัฒนาโปรแกรม ClickRain ดำเนินการตามลำดับการกดปุ่มจริงของผู้ใช้งาน โดยวางแผนให้การใช้งานเป็นไปอย่างราบรื่น ครอบคลุมทั้งการเตรียมข้อมูล การคลิกเลือกอำเภอที่มีฝน การใส่ลูกศรลม และการสร้างอินโฟกราฟิกที่พร้อมใช้งานหรือส่งต่อ โดยมีลำดับการพัฒนา ดังนี้:

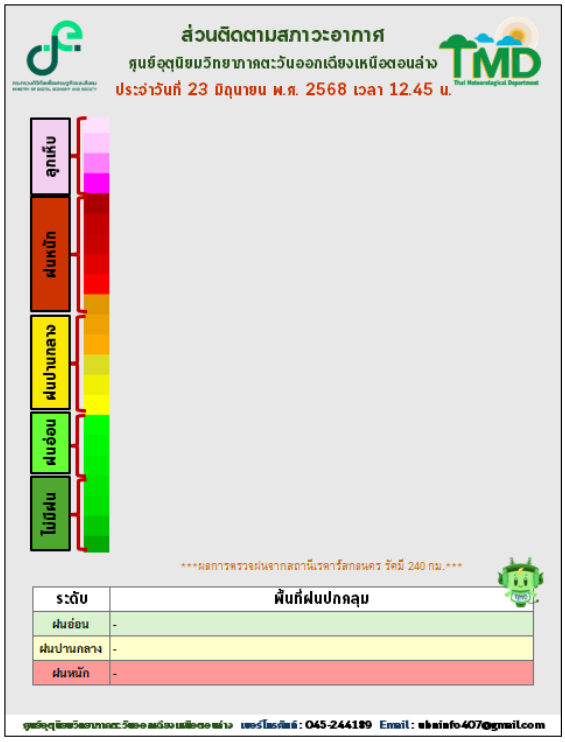
1. **วางแผนลำดับการทำงานของโปรแกรม**

   • ออกแบบลำดับการใช้งานแบบเป็นขั้นตอน (step-by-step) เริ่มตั้งแต่โหลดภาพเรดาร์ ⟶ เลือกอำเภอที่มีฝน ⟶ ใส่ลูกศรลม ⟶ สร้างภาพรายงาน ⟶ คัดลอกภาพเพื่อส่งต่อ

2. **สร้างเทมเพลตภาพอินโฟกราฟิก**

• ออกแบบพื้นที่แสดงผลบน Excel โดยจัดวางตารางข้อมูล ตำแหน่งของภาพเรดาร์ และวางวัตถุรูปร่าง (Shape) ที่ใช้แทนอำเภอต่าง ๆ (Overlay Shape) เพื่อรองรับการคลิก

• ปรับขนาดพื้นที่รายงานให้อยู่ในขอบเขตที่สามารถส่งออกเป็นภาพได้อย่างเหมาะสม



ภาพที่ 8 แสดงการออกแบบพื้นที่แสดงผลบน Excel

3. **โหลดภาพเรดาร์จากเว็บไซต์**

• ใช้ VBA ดึงภาพเรดาร์จาก URL (https://weather.tmd.go.th/skn/skn240_latest.jpg) และแสดงไว้ในตำแหน่ง D7

• มีการปรับขนาด (crop) ให้ภาพพอดีกับพื้นที่อินโฟกราฟิกโดยอัตโนมัติ

4. **กำหนดวันเวลาให้ภาพอินโฟกราฟิกสอดคล้องกับภาพเรดาร์**

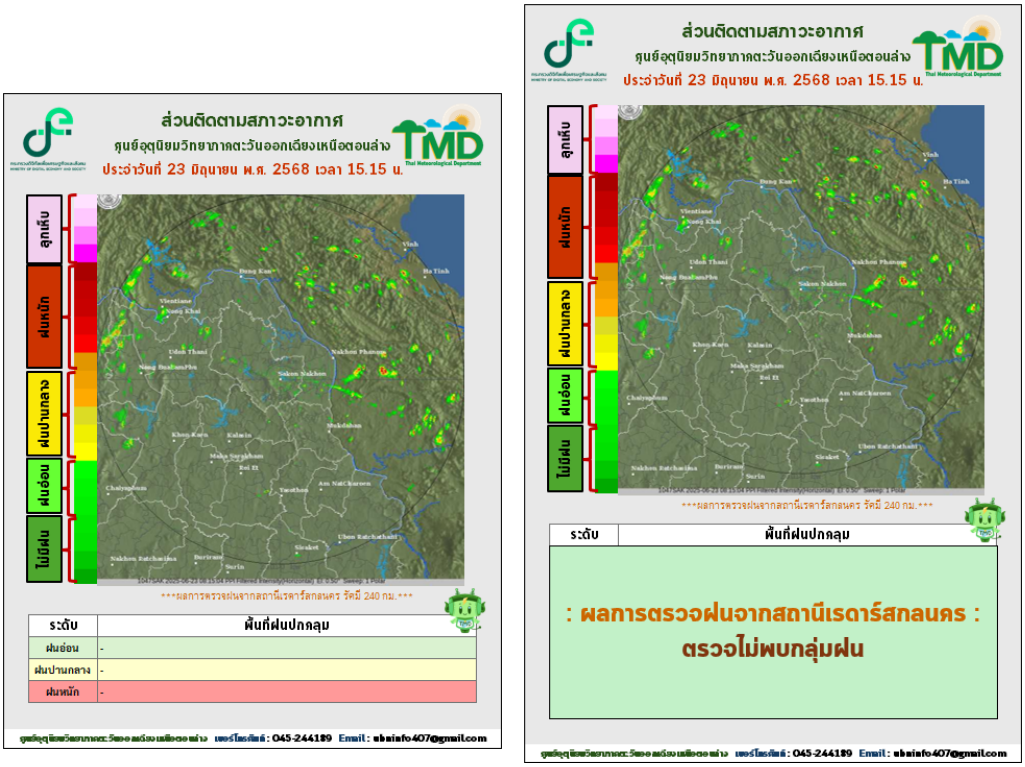
• โปรแกรมใช้สูตรใน Excel เพื่ออัพเดต วันที่และเวลาอัตโนมัติตามเวลาคอมพิวเตอร์

• หากเวลาของภาพเรดาร์ไม่ตรงกับเวลาระบบ ผู้ใช้สามารถปรับเวลาได้โดยใช้ปุ่มบนแถบควบคุมการทำงาน (Control Panel)

5. **เลือกแบบรายงานแบบพบกลุ่มฝน หรือไม่พบกล่มฝน**

• ผู้ใช้เลือกได้ว่าจะรายงานสถานการณ์ “พบกลุ่มฝน” หรือ “ไม่พบกลุ่มฝน” โดยใช้ปุ่มตัวเลือก (OptionButton)

• เทมเพลตของภาพรายงานจะเปลี่ยนไปตามโหมดที่เลือก เช่น สีพื้นหลังหรือข้อความประกอบ



ภาพที่ 9 แสดงการออกแบบเทมเพลตที่แตกต่างกันระหว่าง “พบกลุ่มฝน” หรือ “ไม่พบกลุ่มฝน”

6. **เลือกระดับความรุนแรงฝนและการรายงานฝน**

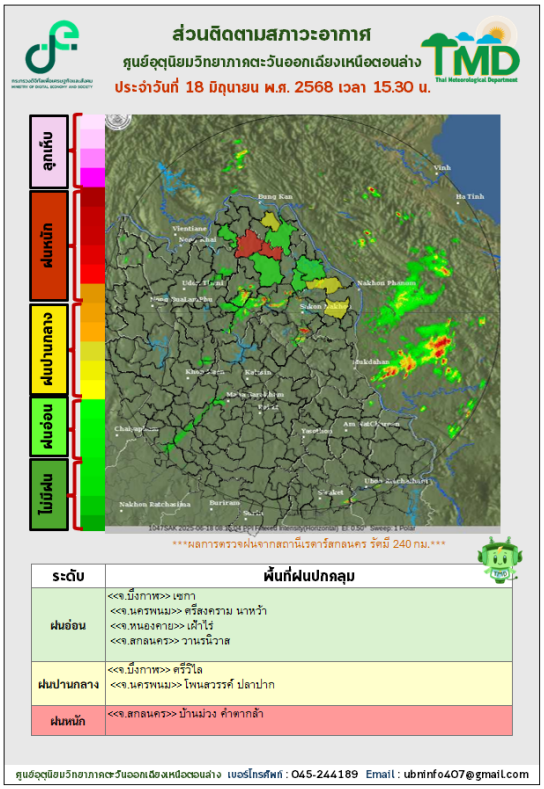
• โปรแกรมรองรับการรายงานฝนแยกตามระดับความรุนแรง ได้แก่ ฝนอ่อน ฝนปานกลาง ฝนหนัก

• เมื่อเลือกระดับฝนด้วยปุ่มตัวเลือก (OptionButton) แล้ว ผู้ใช้สามารถคลิกวัตถุรูปร่าง (Shape) บนอำเภอเพื่อระบุพื้นที่ฝนตามระดับที่เลือก

• โปรแกรมจะบันทึกชื่ออำเภอไว้ในคอลัมน์ที่แยกตามระดับความรุนแรง และแสดงสีของวัตถุรูปร่าง (Shape) ให้แตกต่างกัน

• โปรแกรมใช้ตารางเชื่อมโยงชื่อวัตถุรูปร่าง (Shape) กับชื่ออำเภอ (ในชีตเดียวกัน) เพื่อให้ฟังก์ชัน GetProvinceSummary (Module5) อ่านชื่ออำเภอที่คลิก

• กลไกคลิก shape และการบันทึกค่าจะดำเนินการผ่านโค้ดใน Module6



ภาพที่ 10 แสดงการออกแบบการคลิกเลือกพื้นที่ฝนบนวัตถุรูปร่าง (Shape) ตามระดับความรุนแรงฝน

7. **เรียกดูภาพเรดาร์แบบเคลื่อนไหว**

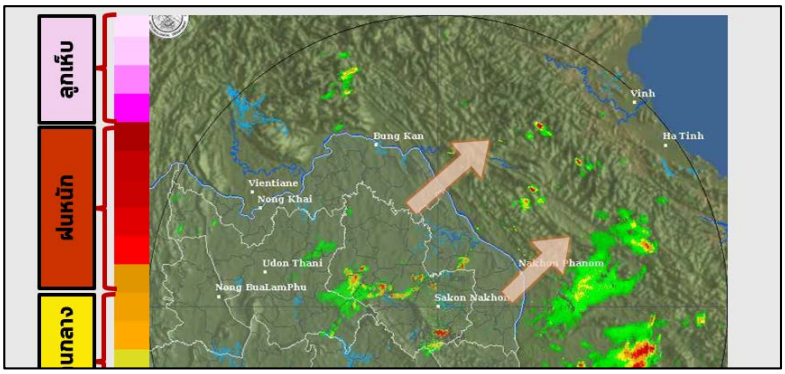
• มีปุ่มสำหรับเปิดลิงค์ภาพเรดาร์แบบเคลื่อนไหว (Radar Loop) บนเว็บ เพื่อช่วยในการวิเคราะห์ทิศทางของกลุ่มฝนก่อนใส่ลูกศร

8. **แทรกลูกศรทิศลมตามการคลิกบนภาพเรดาร์**

• ผู้ใช้คลิก 2 ครั้งบนภาพเพื่อตั้งจุดเริ่มต้นและปลายทางของลูกศร

• โปรแกรมจะวาดเส้นตรงหนา พร้อมหัวลูกศรแบบโปร่งใส และเอฟเฟกต์ Glow เพื่อเน้นทิศทางกลุ่มฝนที่ต้องการรายงาน

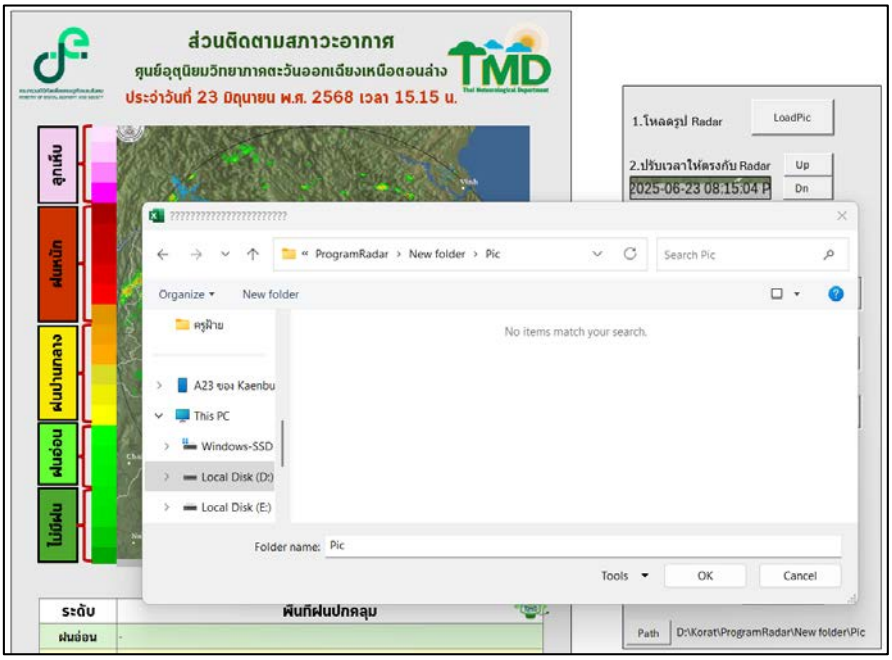
• ลูกศรสามารถคลิกเพื่อลบได้ภายหลัง หากต้องการแก้ไข



ภาพที่ 11 แสดงการออกแบบวาดลูกศรแสดงทิศทางการเคลื่อนที่ของกลุ่มฝนโดยการคลิก 2 ครั้ง

9. **สร้างรูปภาพรายงาน**
   - โปรแกรมสร้าง .ChartObject จากพื้นที่ B2:E16 ซึ่งรวมข้อมูลจากตารางและภาพเรดาร์
   - ส่งออกเป็นภาพ .JPG ไปยัง path ที่ผู้ใช้กำหนดไว้ในเซลล์ H11
   - ตั้งชื่อไฟล์อัตโนมัติตามวันที่และชื่อรายงาน เพื่อสะดวกต่อการจัดเก็บหลายรายการ

10. **เลือกโฟลเดอร์ปลายทาง**
   - ผู้ใช้สามารถคลิกปุ่มเพื่อเลือกโฟลเดอร์ที่ต้องการใช้จัดเก็บภาพ
   - Path ที่เลือกจะถูกบันทึกไว้ในเซลล์ H11 เพื่อใช้งานซ้ำในอนาคต



ภาพที่ 12 แสดงการออกแบบส่วนควบคุมสำหรับให้ผู้ใช้เลือกโฟลเดอร์ปลายทางในการจัดเก็บไฟล์ภาพ

ภาคผนวก

ตารางสรุปการเชื่อมโยง: ปุ่ม ⟶ Module ⟶ ฟังก์ชัน

ลำดับ	ปุ่มควบคุม	ชื่อปุ่มใน VBA	Module	Macro ที่เรียกใช้
1	LoadPic	CommandButton1	Module1	LoadAndCropImageFromURL
			Module2	LoadGIFF
			Module3	ClearFillInCLICK
			Module7	EnableOFF
			Module9	DisableSettings, EnableSettings
2	Up	CommandButton2	Module9	DisableSettings, EnableSettings
3	Dn	CommandButton3	Module9	DisableSettings, EnableSettings
4	พบกลุ่มฝน	OptionButton4	Module9	DisableSettings, EnableSettings
			Module11	hideArrow
5	ไม่พบกลุ่มฝน	OptionButton5	Module9	DisableSettings, EnableSettings
			Module11	hideArrow
6	ประเภทฝน 1 (3.1)	OptionButton1	Module9	DisableSettings, EnableSettings
7	ประเภทฝน 2 (3.2)	OptionButton2	Module9	DisableSettings, EnableSettings
8	ประเภทฝน 3 (3.3)	OptionButton3	Module9	DisableSettings, EnableSettings
9	Radar Loop	ToggleButton1	Module9	DisableSettings, EnableSettings
			Module10	AdjustRowHeight_D11_D13
10	Direction	ToggleButton2	Module11	showArrow, AssignMacroToImage, AssignMacroFromArrowsInD7, RemoveMacroFromImage, RemoveMacroFromArrowsInD7
11	Path	CommandButton4	Module12	SelectFolderAndSavePath

12	Build	CommandButton5	Module9	DisableSettings, EnableSettings
			Module13	SaveRadarInfoAsJpg
			Module15	CopySavedImageToClipboard
13	Shape ขอบเขตอำเภอ	ShapeToText_Click	Module6	ShapeCount_Click, ShapeDel_Click, ClearColor
			Module9	DisableSettings, EnableSettings

1.ปุ่ม ActiveX

1.1 ปุ่ม: LoadPic

ใช้สำหรับโหลดภาพเรดาร์ล่าสุดจากลิงค์ของกรมอุตุฯ พร้อมล้างค่าและรีเซ็ตสถานะของปุ่มและข้อมูลที่เกี่ยวข้องก่อนเริ่มการวิเคราะห์รอบใหม่

```
Private Sub CommandButton1_Click()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Sheet1")
    ws.Unprotect Password:=""
    ' ปิดโหมด Loop หากเปิดค้างไว้
    If Sheet1.ToggleButton1.Value = True Then
        Sheet1.ToggleButton1.Value = False
        ws.Unprotect Password:=""
    End If
    Call Module9.DisableSettings
    ' โหลดภาพเรดาร์จากลิงค์และตัดภาพให้พอดี
    Call Module1.LoadAndCropImageFromURL
    ' โหลดภาพแบบ loop (GIF) มาซ้อน
    Call Module2.LoadGIFF
    ' ล้างค่าที่ใช้ในการควบคุมมุมภาพ
    Sheet1.Range("J5").ClearContents
    ' ซ่อนปุ่มคลิกอำเภอ และรีเซ็ตสถานะของปุ่มต่าง ๆ
    Sheet1.Shapes("CLICK").Visible = msoFalse
```

```
    OptionButton4.Value = False
    OptionButton5.Value = False
    ToggleButton1.Value = False
    ToggleButton2.Value = False
    ' ล้างข้อมูลผลการวิเคราะห์ฝน
    Sheet1.Range("N19:P66").ClearContents
    ' ปิดสถานะของการป้อนข้อมูลอัตโนมัติ
    Call Module7.EnableOFF
    ' ล้างข้อความที่เคยป้อนลงบน shape
    Call Module3.ClearFillInCLICK
    Call Module9.EnableSettings
    ws.Protect Password:=""
End Sub
```

1.2 ปุ่ม: Up

ใช้สำหรับเพิ่มค่ามุมมองของเรดาร์ไปทางขวาทีละ 15 หน่วย โดยมีค่ามากที่สุดคือ +45 เพื่อให้ผู้ใช้งานสามารถเลื่อนตำแหน่งภาพเรดาร์ไปข้างหน้า

```
Private Sub CommandButton2_Click()
    Dim stepSize As Integer
    Dim maxValue As Integer
    Dim TargetCell As Range
    Dim newValue As Integer
    Call Module9.DisableSettings
```

```
    stepSize = 15
    maxValue = 45
    Set TargetCell = ThisWorkbook.Sheets("Sheet1").Range("J5")
    ‘ถ้าเซลล์ J5 ว่างหรือไม่ใช่ตัวเลข ให้เริ่มต้นที่ 0
    If Not IsNumeric(TargetCell.Value) Then
        TargetCell.Value = 0
    End If
    ' เพิ่มค่าทีละ 15 จนถึงค่าสูงสุด
    newValue = TargetCell.Value + stepSize
    If newValue > maxValue Then newValue = maxValue
    TargetCell.Value = newValue
    Call Module9.EnableSettings
End Sub
```

1.3 ปุ่ม: Dn

ใช้สำหรับลดค่ามุมมองของเรดาร์ไปทางซ้ายทีละ 15 หน่วย โดยมีค่าน้อยที่สุดคือ -45 เพื่อให้ผู้ใช้งานสามารถเลื่อนตำแหน่งภาพเรดาร์ไปข้างหลัง

```
Private Sub CommandButton3_Click()
    Dim stepSize As Integer
    Dim minValue As Integer
    Dim TargetCell As Range
    Dim newValue As Integer
    stepSize = 15
```

```
    minValue = -45
    Set TargetCell = ThisWorkbook.Sheets("Sheet1").Range("J5")
    ' ถ้าเซลล์ J5 ว่างหรือไม่ใช่ตัวเลข ให้เริ่มต้นที่ 0
    If Not IsNumeric(TargetCell.Value) Then
        TargetCell.Value = 0
    End If
    ' ลดค่าทีละ 15 จนถึงค่าต่ำสุด
    newValue = TargetCell.Value - stepSize
    If newValue < minValue Then newValue = minValue
    TargetCell.Value = newValue
End Sub
```

1.4 ปุ่ม: พบกลุ่มฝน (OptionButton4)

เมื่อตรวจพบกลุ่มฝน ผู้ใช้งานสามารถคลิกที่อำเภอเพื่อแสดงชื่อบนพื้นที่วิเคราะห์ โดยจะเปิดการใช้งานตัวเลือกของอำเภอและคอมโบเพื่อใส่ข้อความกำกับเพิ่มเติม

```
Private Sub OptionButton4_Click()
    Call Module9.DisableSettings
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Sheet1")
    ws.Unprotect Password:=""
    ' ซ่อนลูกศรที่อาจมีอยู่ก่อนหน้า
    Call Module11.hideArrow
    If OptionButton4.Value = True Then
```

```
        Sheet1.Shapes("CLICK").Visible = msoTrue ' เปิดพื้นที่คลิกเพื่อใส่ข้อความ
        OptionButton1.Enabled = True
        OptionButton2.Enabled = True
        OptionButton3.Enabled = True
        OptionButton1.Value = True
        ComboBox1.Enabled = True
    End If
    ' ซ่อนภาพข้อความ "ไม่พบกลุ่มฝน"
    Sheet1.Shapes("Nodetect").Visible = msoFalse
    ActiveWindow.ScrollRow = 7
    Rows("11:13").AutoFit
    ws.Protect Password:=""
    Call Module9.EnableSettings
End Sub
```

1.5 ปุ่ม: ไม่พบกลุ่มฝน (OptionButton5)

เมื่อตรวจไม่พบกลุ่มฝน ระบบจะซ่อนพื้นที่สำหรับคลิกแสดงอำเภอ และแสดงภาพข้อความ “ไม่พบกลุ่มฝน” แทน เพื่อให้ภาพสรุปมีความชัดเจน

```
Private Sub OptionButton5_Click()
    Call Module9.DisableSettings
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Sheet1")
    ws.Unprotect Password:=""
```

```
    If OptionButton5.Value = True Then
        Sheet1.Shapes("CLICK").Visible = msoFalse ' ซ่อนพื้นที่พิมพ์ข้อความ
        Call Module11.hideArrow
        OptionButton1.Enabled = False
        OptionButton2.Enabled = False
        OptionButton3.Enabled = False
        ComboBox1.Enabled = False
        ComboBox2.Enabled = False
        ComboBox3.Enabled = False
    End If
    ' ปรับขนาดและแสดงข้อความ “ไม่พบกลุ่มฝน”
    With Sheet1.Shapes("Nodetect")
        .LockAspectRatio = msoFalse
        .Width = 456.8
        .Height = 180.6
        .Visible = msoTrue
    End With
    ActiveWindow.ScrollRow = 7
    Rows("11:13").RowHeight = 60
    ws.Protect Password:=""
    Call Module9.EnableSettings
End Sub
```

1.6 ปุ่ม 3.1 พบกลุ่มฝน (OptionButton1)

เมื่อเลือกปุ่มนี้ ระบบจะเปิดใช้งาน ComboBox1 สำหรับใส่คำอธิบายเพิ่มเติมเกี่ยวกับฝนเล็กน้อยหรืออื่นๆ และปิดการใช้งานกล่องอื่นที่ไม่เกี่ยวข้อง

```
Private Sub OptionButton1_Click()
    Call Module9.DisableSettings
    If OptionButton1.Value = True Then
        ComboBox1.Enabled = True
    End If
    ComboBox2.Enabled = False
    ComboBox3.Enabled = False
    Call Module9.EnableSettings
End Sub
```

1.7 ปุ่ม 3.2 พบกลุ่มฝน (OptionButton2)

ระบบจะเปิดใช้งาน ComboBox2 เพื่อใส่รายละเอียดกรณีพบฝนระดับปานกลางหรืออื่นๆ และปิด ComboBox อื่น

```
Private Sub OptionButton2_Click()
    Call Module9.DisableSettings
    If OptionButton2.Value = True Then
        ComboBox2.Enabled = True
    End If
    ComboBox1.Enabled = False
    ComboBox3.Enabled = False
    Call Module9.EnableSettings
    End Sub
```

1.8 ปุ่ม 3.3 พบกลุ่มฝน (OptionButton3)

เปิด ComboBox3 สำหรับพิมพ์รายละเอียดฝนรุนแรงหรืออื่นๆ พร้อมปิดการทำงานของช่องอื่น

```
Private Sub OptionButton3_Click()
    Call Module9.DisableSettings
    If OptionButton3.Value = True Then
        ComboBox3.Enabled = True
    End If
    ComboBox1.Enabled = False
    ComboBox2.Enabled = False
    Call Module9.EnableSettings
End Sub
```

1.9 ปุ่ม: แสดง/ซ่อนเรดาร์ลูป (Radar Loop – ToggleButton1)

ปุ่มนี้ทำหน้าที่เปิดหรือปิดการแสดงผลของภาพเรดาร์แบบเคลื่อนไหว (GIF) บนแผ่นงาน โดยจะปรับตำแหน่งภาพให้แสดงอยู่ด้านบนสุดเมื่อต้องการให้แสดง และส่งไปด้านหลังเมื่อปิด

```
Private Sub ToggleButton1_Click()
    Dim ws As Worksheet
    Dim shp As Shape
    Call Module9.DisableSettings
    Set ws = ThisWorkbook.Sheets("Sheet1")
    ws.Unprotect Password:=""
    ' ปิดฟีเจอร์อื่นก่อนแสดงเรดาร์ลูป
```

```
Sheet1.Shapes("CLICK").Visible = msoFalse
    OptionButton4.Value = False
    OptionButton5.Value = False
    OptionButton1.Enabled = False
    OptionButton2.Enabled = False
    OptionButton3.Enabled = False
    ComboBox1.Enabled = False
    ComboBox2.Enabled = False
    ComboBox3.Enabled = False
    ' แสดงภาพเรดาร์ลูป (GIF) ถ้าเปิด Toggle
    If ToggleButton1.Value = True Then
        ws.Shapes("LoopGIFF").ZOrder msoBringToFront ' นำภาพไปไว้ด้านบนสุด
        ws.Shapes("LoopGIFF").Visible = msoTrue
        Call Module10.AdjustRowHeight_D11_D13
        Sheet1.ToggleButton2.Value = False
        OptionButton4.Enabled = False
        OptionButton5.Enabled = False
    Else
        ' ซ่อนภาพเรดาร์ลูป หากปิด Toggle
        ws.Shapes("LoopGIFF").ZOrder msoSendToBack
        ws.Shapes("LoopGIFF").Visible = msoFalse
        OptionButton4.Enabled = True
```

```
        OptionButton5.Enabled = True
    End If
    ws.Protect Password:=""
    Call Module9.EnableSettings
End Sub
```

1.10 ปุ่ม: ทิศทางกลุ่มฝน (Direction – ToggleButton2)

ปุ่มนี้ทำหน้าที่เปิดหรือปิดการแสดงลูกศรทิศทางของกลุ่มฝน โดยเมื่อเปิดจะให้ผู้ใช้สามารถคลิกเพื่อวาดลูกศร (เริ่มต้นจากคลิกแรกและสิ้นสุดที่คลิกที่สอง) และเมื่อปิดจะลบมาโครที่ผูกกับรูปภาพออก เพื่อไม่ให้สามารถวาดหรือลบลูกศรได้อีก

```
Private Sub ToggleButton2_Click()
    Sheet1.Shapes("CLICK").Visible = msoFalse
    Call Module11.showArrow ' แสดงลูกศรที่มีอยู่
    ' ปิดปุ่มอื่นที่เกี่ยวข้อง
    OptionButton4.Value = False
    OptionButton5.Value = False
    OptionButton1.Enabled = False
    OptionButton2.Enabled = False
    OptionButton3.Enabled = False
    ComboBox1.Enabled = False
    ComboBox2.Enabled = False
    ComboBox3.Enabled = False
    ' ปิด Radar Loop หากเปิดอยู่
    If ToggleButton2.Value = True Then ToggleButton1.Value = False
```

```
    If ToggleButton2.Value = True Then
        Sheet1.Shapes("CLICK").Visible = msoFalse ' ซ่อนกล่องพิมพ์ข้อความ
        Call Module11.AssignMacroToImage ' ผูกมาโครให้ภาพเพื่อตรวจจับการคลิก
        Call Module11.AssignMacroFromArrowsInD7 ' ผูกมาโครให้ลูกศรในเซลล์ D7
        OptionButton4.Enabled = False
        OptionButton5.Enabled = False
    ElseIf ToggleButton2.Value = False Then
        Call Module11.RemoveMacroFromImage ' เอามาโครออกจากภาพ
        Call Module11.RemoveMacroFromArrowsInD7 ' เอามาโครออกจากลูกศร
        OptionButton4.Enabled = True
        OptionButton5.Enabled = True
    End If
End Sub
```

1.11 ปุ่ม: เลือกโฟลเดอร์ปลายทาง (Path – CommandButton4)

ปุ่มนี้ใช้สำหรับให้ผู้ใช้เลือกโฟลเดอร์ที่จะใช้เป็นตำแหน่งปลายทางในการบันทึกภาพผลวิเคราะห์เรดาร์ที่ถูกสร้างขึ้น เช่น ภาพ JPG ที่ใช้ในการรายงานหรือเผยแพร่ต่อไป

```
Private Sub CommandButton4_Click()
    Call Module12.SelectFolderAndSavePath
End Sub
```

คำอธิบายเพิ่มเติม:

คำสั่ง Module12.SelectFolderAndSavePath จะเรียกฟังก์ชันจาก Module12 เพื่อเปิดหน้าต่างให้ผู้ใช้เลือกโฟลเดอร์

โฟลเดอร์ที่เลือกจะถูกจดจำไว้เพื่อนำไปใช้โดยอัตโนมัติในขั้นตอนการบันทึกภาพเรดาร์ในปุ่มถัดไป (เช่น CommandButton5)

1.12 ปุ่ม: สร้างภาพ (Build – CommandButton5)

ปุ่มนี้ทำหน้าที่สร้างภาพผลวิเคราะห์เรดาร์จากข้อมูลที่ผู้ใช้จัดวางไว้ และทำการคัดลอกภาพนั้นลงใน Clipboard เพื่อสะดวกต่อการนำไปวางในแอปอื่น เช่น PowerPoint, Word หรือ Line รวมถึงใช้เผยแพร่ในรายงานต่างๆ ได้ทันที

```
Private Sub CommandButton5_Click()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Sheet1")
    ws.Unprotect Password:=""
    Call Module9.DisableSettings
    Call Module13.SaveRadarInfoAsJpg  ' บันทึกภาพผลวิเคราะห์เป็น JPG
    Call Module15.CopySavedImageToClipboard  ' คัดลอกภาพไปยัง Clipboard
    ws.Protect Password:=""
    Call Module9.EnableSettings
End Sub
```

1.13 ปุ่ม: คลิก Shape ขอบเขตอำเภอ (Shape Overlay)

(เรียกจากชื่อ Shape โดยตรง ไม่ใช่ Control)

```
' จัดการคลิกที่ shape เพื่อเปลี่ยนสีตามชนิดฝน และบันทึก/ลบข้อมูล
Sub ShapeToText_Click()
    Dim shp As Shape
    Dim ws As Worksheet
    Dim r As Long, g As Long, b As Long
```

```
Dim optIndex As Integer
Dim selectedIndex As Integer
Call Module9.DisableSettings
Set ws = ThisWorkbook.Sheets("Sheet1")
ws.Unprotect Password:=""
On Error Resume Next
Set shp = ws.Shapes(Application.Caller)
On Error GoTo 0
If Not shp Is Nothing Then
    ' อ่าน index ของประเภทฝน และระดับฝนที่ผู้ใช้เลือก
    If Sheet1.OptionButton1.Value Then
        optIndex = 1: selectedIndex = Sheet1.ComboBox1.ListIndex
    ElseIf Sheet1.OptionButton2.Value Then
        optIndex = 2: selectedIndex = Sheet1.ComboBox2.ListIndex
    ElseIf Sheet1.OptionButton3.Value Then
        optIndex = 3: selectedIndex = Sheet1.ComboBox3.ListIndex
    End If
    ' กำหนดค่าสี RGB สำหรับระดับฝน
    Dim colorsRGB(0 To 5, 0 To 2) As Long
    colorsRGB(0, 0) = 255: colorsRGB(0, 1) = 255: colorsRGB(0, 2) = 255 ' ขาว (ล้าง)
    colorsRGB(1, 0) = 0:   colorsRGB(1, 1) = 255: colorsRGB(1, 2) = 0   ' เขียวอ่อน
    colorsRGB(2, 0) = 255: colorsRGB(2, 1) = 255: colorsRGB(2, 2) = 0   ' เหลือง
```

```
    colorsRGB(3, 0) = 255: colorsRGB(3, 1) = 0:   colorsRGB(3, 2) = 0   ' แดง
    colorsRGB(4, 0) = 180: colorsRGB(4, 1) = 255: colorsRGB(4, 2) = 0   ' เขียวอ่อนเข้ม
    colorsRGB(5, 0) = 255: colorsRGB(5, 1) = 165: colorsRGB(5, 2) = 0   ' ส้ม
    r = colorsRGB(selectedIndex, 0)
    g = colorsRGB(selectedIndex, 1)
    b = colorsRGB(selectedIndex, 2)
    ' หาก shape มีสีเดิมแล้ว ⟶ ล้างสีและลบข้อมูล
    If shp.Fill.Transparency <> 1 Then
        If shp.Fill.ForeColor.RGB = RGB(r, g, b) Then
            shp.Fill.Transparency = 1
            shp.Line.ForeColor.RGB = RGB(0, 0, 0)
            Call Module6.ShapeDel_Click
        End If
    Else
        ' หากยังไม่มีสี ⟶ เติมสีและบันทึก
        shp.Fill.Visible = msoTrue
        shp.Fill.Transparency = 0.5
        shp.Fill.ForeColor.RGB = RGB(r, g, b)
        Call Module6.ShapeCount_Click
    End If
End If
Set shp = Nothing
```

```
    ws.Protect Password:=""
    Call Module9.EnableSettings
End Sub
```

2. Module

2.1 Module1

2.1.1 ชื่อฟังก์ชัน LoadAndCropImageFromURL

```
Sub LoadAndCropImageFromURL()
    Dim ws As Worksheet
    Dim imageURL As String
    Dim rng As Range
    Dim shp As Shape
    Dim pic As Shape
    Set ws = ThisWorkbook.Sheets("Sheet1")
    imageURL = "https://weather.tmd.go.th/skn/skn240_latest.jpg"
    Set rng = ws.Range("D7")
    ' ลบภาพทั้งหมดที่อยู่ทับเซลล์ D7 เพื่อเคลียร์ก่อนวางภาพใหม่
    For Each shp In ws.Shapes
        If Not Intersect(shp.TopLeftCell, rng) Is Nothing Then
            shp.Delete
        End If
    Next shp
```

```
' แทรกภาพจาก URL เข้าสู่ชีต
Dim insertedPic As Picture
Set insertedPic = ws.Pictures.Insert(imageURL)
' ถ้าแทรกภาพสำเร็จ ให้ครอปและปรับขนาดตำแหน่งของภาพให้พอดีเซลล์
If Not insertedPic Is Nothing Then
    Set pic = insertedPic.ShapeRange(1)
    With pic.PictureFormat
        .CropLeft = 5400
        .CropTop = 2100
    End With
    With pic
        .LockAspectRatio = msoTrue
        .Left = rng.Left
        .Top = rng.Top
        .Width = rng.Width
        .Height = rng.Height
    End With
    pic.Name = "PicRadar"
    pic.ZOrder msoSendToBack
Else
    MsgBox "ไม่สามารถโหลดภาพเรดาร์ได้", vbExclamation
End If
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน LoadAndCropImageFromURL ทำหน้าที่โหลดภาพเรดาร์ล่าสุดจากเว็บไซต์ของกรมอุตุนิยมวิทยา (ผ่าน URL แบบคงที่) และวางภาพนี้ลงที่เซลล์ D7 ใน Sheet1 โดยก่อนวางจะลบภาพเดิมที่อาจอยู่ทับ D7 ออกก่อน จากนั้นจะครอปภาพให้พอดีเฉพาะส่วนที่ต้องการ พร้อมปรับขนาดและวางไว้ในตำแหน่งที่เหมาะสม หากโหลดไม่สำเร็จจะแสดงข้อความแจ้งเตือนผู้ใช้

2.2 Module2

2.2.1 ชื่อฟังก์ชัน LoadGIFF

```
Sub LoadGIFF()
    Dim ws As Worksheet
    Dim imageURL As String
    Dim rng As Range
    Dim shp As Shape
    Dim pic As Shape
    Set ws = ThisWorkbook.Sheets("Sheet1")
    imageURL = "https://weather.tmd.go.th/skn/skn240Loop.gif"
    Set rng = ws.Range("D7")
    ' แทรกรูปภาพ GIF จาก URL เข้าสู่ชีต
    Dim insertedPic As Picture
    Set insertedPic = ws.Pictures.Insert(imageURL)
    ' ถ้าแทรกรูปภาพสำเร็จ จะทำการปรับขนาดและครอปตำแหน่งของภาพ
    If Not insertedPic Is Nothing Then
        Set pic = insertedPic.ShapeRange(1)
        With pic
```

```
            .LockAspectRatio = msoFalse
            .Left = rng.Left
            .Top = rng.Top
            .Width = pic.Width * 83 / 100
            .Height = pic.Height * 83 / 100
        End With
        With pic.PictureFormat
            .Crop.PictureOffsetX = -39
            .Crop.PictureOffsetY = -15
        End With
        pic.ZOrder msoSendToBack
        pic.Name = "LoopGIFF"
        pic.Visible = msoFalse
    Else
        MsgBox "ไม่สามารถโหลดรูปภาพได้", vbExclamation
    End If
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน LoadGIFF ใช้สำหรับโหลดภาพเรดาร์แบบเคลื่อนไหว (GIF) จาก URL ที่กำหนด และวางลงในเซลล์ D7 ของ Sheet1 โดยไม่มีการลบภาพเก่าออก (บรรทัดที่ลบถูกคอมเมนต์ไว้) เมื่อโหลดภาพได้สำเร็จ ฟังก์ชันจะทำการปรับขนาดและเลื่อนตำแหน่งของภาพเล็กน้อยเพื่อให้แสดงผลได้เหมาะสม พร้อมทั้งตั้งชื่อภาพว่า LoopGIFF และซ่อนไว้ในเบื้องต้นเพื่อให้แสดงเมื่อผู้ใช้กดปุ่ม ToggleButton ที่เกี่ยวข้อง

2.3 Module3

2.3.1 ชื่อฟังก์ชัน ClearFillInCLICK

```
Sub ClearFillInCLICK()
    Dim ws As Worksheet
    Dim shp As Shape
    Dim grp As Shape
    Set ws = ThisWorkbook.Sheets("Sheet1")
    On Error Resume Next
    Set grp = ws.Shapes("CLICK")
    On Error GoTo 0
    If Not grp Is Nothing Then
        For Each shp In grp.GroupItems
            shp.Fill.Visible = msoTrue ' เปิดการแสดงผลการเติมสี
            shp.Fill.Transparency = 1 ' ล้างการเติมสีให้โปร่งใส
            shp.Line.ForeColor.RGB = RGB(0, 0, 0) ' ขอบสีดำ
        Next shp
    Else
        MsgBox "ไม่พบ CLICK ในชีตนี้", vbExclamation, "ข้อผิดพลาด"
    End If
    Set shp = Nothing
    Set grp = Nothing
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน ClearFillInCLICK มีหน้าที่ล้างการระบายสีของวัตถุในกลุ่ม CLICK บนชีต Sheet1 โดยจะทำให้วัตถุในกลุ่มโปร่งใสและมีขอบสีดำ เพื่อใช้แสดงผลใหม่ในรอบถัดไป หากไม่พบวัตถุกลุ่ม CLICK ระบบจะแจ้งเตือน

2.4 Module7

2.4.1 ชื่อฟังก์ชัน EnableOFF

```
Sub EnableOFF()
Sheet1.OptionButton1.Enabled = False ' ปิดการใช้งานปุ่มประเภทฝน 3.1
Sheet1.ComboBox1.Enabled = False ' ปิดการใช้งาน ComboBox ของประเภทฝน 3.1
Sheet1.OptionButton2.Enabled = False ' ปิดการใช้งานปุ่มประเภทฝน 3.2
Sheet1.ComboBox2.Enabled = False ' ปิดการใช้งาน ComboBox ของประเภทฝน 3.2
Sheet1.OptionButton3.Enabled = False ' ปิดการใช้งานปุ่มประเภทฝน 3.3
Sheet1.ComboBox3.Enabled = False ' ปิดการใช้งาน ComboBox ของประเภทฝน 3.3
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน EnableOFF ทำหน้าที่ปิดการใช้งาน (Disable) ปุ่มตัวเลือกประเภทฝนทั้ง 3 ปุ่ม (3.1, 3.2, 3.3) และ ComboBox ที่เกี่ยวข้องทั้งหมด เพื่อป้องกันไม่ให้ผู้ใช้เลือกประเภทฝนในบางกรณี เช่น เมื่อเปิดภาพเรดาร์แบบ Loop หรือในกรณีที่ไม่พบกลุ่มฝน

2.5 Module9

2.5.1 ชื่อฟังก์ชัน DisableSettings

```
Sub DisableSettings()
Application.ScreenUpdating = False ' ปิดการอัปเดตหน้าจอขณะรันโค้ดเพื่อเพิ่มความเร็ว
```

Application.Calculation = xlCalculationManual ' ปิดการคำนวณอัตโนมัติเพื่อป้องกันการคำนวณซ้ำหลายครั้ง

Application.EnableEvents = False ' ปิดการทำงานของ Event ต่างๆ เช่น Worksheet_Change

End Sub

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน DisableSettings ใช้สำหรับปรับสภาพแวดล้อมของ Excel ให้เหมาะสมก่อนรันโค้ดโดยปิดฟีเจอร์ที่ทำให้การทำงานช้าลง เช่น การอัปเดตหน้าจอ การคำนวณอัตโนมัติ และ Event ต่างๆ เพื่อให้การทำงานของ VBA มีประสิทธิภาพสูงสุด

2.5.2 ชื่อฟังก์ชัน EnableSettings

Sub EnableSettings()

Application.ScreenUpdating = True ' เปิดการอัปเดตหน้าจอกลับมา

Application.Calculation = xlCalculationAutomatic ' เปิดการคำนวณอัตโนมัติกลับคืน

Application.EnableEvents = True ' เปิดการทำงานของ Event ให้กลับมาเป็นปกติ

End Sub

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน EnableSettings ใช้คืนค่าการตั้งค่าที่ถูกปิดไว้ใน DisableSettings ให้กลับสู่สภาวะปกติ เช่น เปิดการอัปเดตหน้าจอ การคำนวณ และ Event ต่างๆ เพื่อให้ผู้ใช้สามารถโต้ตอบกับ Excel ได้ตามปกติหลังจากโค้ดหลักรันเสร็จ

2.6 Module11

2.6.1 ชื่อฟังก์ชัน hideArrow

Sub hideArrow()

' ซ่อนลูกศรทั้งหมดที่ทับอยู่บนเซลล์ D7

```
Dim shp As Shape
Dim ws As Worksheet
Dim TargetCell As Range
' กำหนดชีตและเซลล์เป้าหมาย
Set ws = ThisWorkbook.Sheets("Sheet1")
Set TargetCell = ws.Range("D7")
' วนลูปตรวจสอบวัตถุทั้งหมดในชีต
For Each shp In ws.Shapes
    ' ตรวจเฉพาะวัตถุที่เป็น Connector (เช่นลูกศร)
 If shp.Connector Then
        ' ตรวจชื่อว่าขึ้นต้นด้วย "Arrow" ซึ่งเป็นชื่อที่เราตั้งให้ลูกศร
   If shp.Name Like "Arrow*" Then
            ' ตรวจตำแหน่งว่าทับกับเซลล์ D7 หรือไม่
     If shp.Left >= TargetCell.Left And shp.Left <= TargetCell.Left + TargetCell.Width
And _
        shp.Top >= TargetCell.Top And shp.Top <= TargetCell.Top + TargetCell.Height
Then
                ' ซ่อนลูกศรนี้
                shp.Visible = msoFalse
            End If
        End If
    End If
Next shp
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน showArrow ใช้สำหรับ **แสดงลูกศร (Arrow)** ที่ซ่อนอยู่และมีตำแหน่งทับกับเซลล์ D7 ในชีต Sheet1 ซึ่งเป็นฟังก์ชันทำงานตรงข้ามกับ hideArrow

หลักการทำงาน:

- ระบุตำแหน่งของเซลล์ D7
- ตรวจสอบว่าเป็น Connector และมีชื่อขึ้นต้นว่า "Arrow"
- หากตำแหน่งลูกศรทับ D7 ให้ Visible = msoTrue เพื่อให้ลูกศรแสดงกลับคืน

2.6.2 ชื่อฟังก์ชัน showArrow

```
Sub showArrow()

' แสดงลูกศรทั้งหมดที่อยู่บนเซลล์ D7

Dim shp As Shape

Dim ws As Worksheet

Dim TargetCell As Range

' กำหนดชีตและเซลล์เป้าหมาย

Set ws = ThisWorkbook.Sheets("Sheet1")

Set TargetCell = ws.Range("D7")

' วนลูปตรวจสอบวัตถุทั้งหมดในชีต

For Each shp In ws.Shapes

    ' ตรวจเฉพาะวัตถุที่เป็น Connector (เช่นลูกศร)

    If shp.Connector Then

        ' ตรวจชื่อว่าขึ้นต้นด้วย "Arrow" ซึ่งเป็นชื่อที่เราตั้งให้ลูกศร
```

```
        If shp.Name Like "Arrow*" Then
            ' ตรวจตำแหน่งว่าทับกับเซลล์ D7 หรือไม่
        If shp.Left >= TargetCell.Left And shp.Left <= TargetCell.Left + TargetCell.Width And
_
           shp.Top >= TargetCell.Top And shp.Top <= TargetCell.Top + TargetCell.Height
Then
                ' แสดงลูกศรนี้
                shp.Visible = msoTrue
           End If
        End If
     End If
    Next shp
    End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน showArrow ใช้สำหรับ **แสดงลูกศร (Arrow)** ที่ซ่อนอยู่และมีตำแหน่งทับกับเซลล์ D7 ในชีต Sheet1 ซึ่งเป็นฟังก์ชันตรงข้ามกับ hideArrow

หลักการทำงาน:

- ระบุตำแหน่งของเซลล์ D7
- ตรวจสอบว่าเป็น Connector และมีชื่อขึ้นต้นว่า "Arrow"
- หากตำแหน่งลูกศรทับ D7 ให้ Visible = msoTrue เพื่อให้ลูกศรแสดงกลับคืน

2.6.3 ชื่อฟังก์ชัน AssignMacroToImage

```
Sub AssignMacroToImage()
    Dim shp As Shape
    Dim MacroName As String
    MacroName = "Draw_Thick_Line_Arrow" ' ตั้งชื่อมาโครที่จะเชื่อมกับภาพ
    On Error Resume Next ' ปิดการแสดง error ชั่วคราวในกรณีไม่พบรูป
    Set shp = ActiveSheet.Shapes("PicRadar") ' พยายามค้นหารูปภาพชื่อ "PicRadar" ในชีตปัจจุบัน
    On Error GoTo 0 ' เปิดการแสดง error กลับมา
    If Not shp Is Nothing Then
        shp.OnAction = MacroName
    Else
        MsgBox "ไม่พบรูปภาพชื่อ 'PicRadar' ในชีตนี้!", vbExclamation, "ผิดพลาด"
    End If
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน AssignMacroToImage ใช้สำหรับกำหนดให้รูปภาพที่ชื่อว่า “PicRadar” ทำงานร่วมกับมาโคร Draw_Thick_Line_Arrow เมื่อผู้ใช้คลิกที่ภาพนั้น จะเรียกใช้มาโครเพื่อเริ่มต้นการวาดลูกศร หากไม่มีรูปภาพที่ระบุไว้ในชีต จะมีการแจ้งเตือนผ่านกล่องข้อความเพื่อบอกผู้ใช้ว่ารูปภาพไม่พบ

2.6.4 ชื่อฟังก์ชัน RemoveMacroFromArrowsInD7

```
Sub RemoveMacroFromArrowsInD7()
    Dim shp As Shape
    Dim ws As Worksheet
    Dim TargetCell As Range
    Set ws = ActiveSheet ' ใช้งานกับชีตที่กำลังเปิดอยู่
    Set TargetCell = ws.Range("D7") ' กำหนดเซลล์เป้าหมาย D7
    For Each shp In ws.Shapes ' วนลูปรูปทรงทั้งหมดบนชีต
        If shp.Connector Then ' ตรวจว่าเป็นเส้นเชื่อม (ลูกศร)
            If shp.Name Like "Arrow*" Then ' ตรวจชื่อว่าขึ้นต้นด้วย Arrow
                If shp.Left >= TargetCell.Left And shp.Left <= TargetCell.Left + TargetCell.Width
And _
                   shp.Top >= TargetCell.Top And shp.Top <= TargetCell.Top +
TargetCell.Height Then
                    shp.OnAction = "" ' ลบการกำหนดมาโครออกจากลูกศรที่อยู่ในเซลล์ D7
                End If
            End If
        End If
    Next shp
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน RemoveMacroFromArrowsInD7 ใช้สำหรับลบการเชื่อมโยงมาโครที่ถูกกำหนดให้กับลูกศรทั้งหมดที่อยู่ในเซลล์ D7 โดยตรวจสอบว่าเป็นรูปทรงแบบ Connector (เส้นลูกศร) และมีชื่อขึ้นต้นว่า

"Arrow" เท่านั้น หากอยู่ในพื้นที่ของเซลล์ D7 ก็จะลบมาโครที่เชื่อมโยงออกทันที เพื่อป้องกันการทำงานผิดพลาดหรือไม่ต้องการให้คลิกแล้วเกิดการทำงาน

2.6.5 ชื่อฟังก์ชัน RemoveMacroFromImage

```
Sub RemoveMacroFromImage()
    Dim shp As Shape
    On Error Resume Next
    Set shp = ActiveSheet.Shapes("PicRadar") ' ตรวจสอบว่ามีรูปภาพชื่อ "PicRadar" หรือไม่
    On Error GoTo 0
    If Not shp Is Nothing Then
        shp.OnAction = "" ' ลบมาโครที่ผูกไว้กับรูปภาพออก
    Else
        MsgBox "ไม่พบรูปภาพชื่อ 'PicRadar' ในชีตนี้!", vbExclamation, "ผิดพลาด"
    End If
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน RemoveMacroFromImage ใช้สำหรับลบการกำหนดมาโครออกจากรูปภาพที่มีชื่อว่า "PicRadar" บนชีตปัจจุบัน โดยไม่ต้องลบรูปภาพออก หากไม่พบรูปดังกล่าวจะมีการแจ้งเตือนด้วยกล่องข้อความ เพื่อแจ้งให้ผู้ใช้ทราบว่าไม่พบวัตถุตามที่ต้องการ

2.6.6 ชื่อฟังก์ชัน RemoveMacroFromArrowsInD7

```
Sub RemoveMacroFromArrowsInD7()
    Dim shp As Shape
    Dim ws As Worksheet
    Dim TargetCell As Range
```

```
    Set ws = ActiveSheet ' ใช้ชีตปัจจุบัน
    Set TargetCell = ws.Range("D7") ' กำหนดตำแหน่งเซลล์ D7 ที่ใช้ตรวจสอบ
    For Each shp In ws.Shapes ' วนลูปผ่านทุกรูปร่างในชีต
        If shp.Connector Then ' ถ้าเป็นเส้นเชื่อม (ลูกศร)
            If shp.Name Like "Arrow*" Then ' ถ้าชื่อลูกศรขึ้นต้นด้วย "Arrow"
                If shp.Left >= TargetCell.Left And shp.Left <= TargetCell.Left +
TargetCell.Width And _
                   shp.Top >= TargetCell.Top And shp.Top <= TargetCell.Top +
TargetCell.Height Then
                    shp.OnAction = "" ' ลบมาโครที่ผูกกับลูกศรออก
                End If
            End If
        End If
    Next shp
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน RemoveMacroFromArrowsInD7 จะลบการผูกมาโครออกจากลูกศรทุกเส้นที่ชื่อขึ้นต้นด้วย "Arrow" และวางอยู่ภายในบริเวณเซลล์ D7 ของชีตปัจจุบัน ใช้ในกรณีที่ผู้ใช้ต้องการยกเลิกการทำงานของลูกศรที่คลิกแล้วจะเรียกมาโคร เช่น สำหรับล้างการโต้ตอบกับผู้ใช้

2.7 Module10

2.7.1 ชื่อฟังก์ชัน AdjustRowHeight_D11_D13

```
Sub AdjustRowHeight_D11_D13()
    Dim rowNum As Integer
```

```
    Rows("11:13").AutoFit ' เริ่มต้นปรับความสูงอัตโนมัติให้แถว 11 ถึง 13
    For rowNum = 11 To 13 ' วนลูปผ่านแถว 11 ถึง 13
        If Rows(rowNum).RowHeight <= 60 Then ' ถ้าความสูงน้อยกว่าหรือเท่ากับ 60
            Rows(rowNum).RowHeight = 60 ' กำหนดความสูงขั้นต่ำให้เป็น 60
        Else
            Rows(rowNum).AutoFit ' ถ้ามากกว่า 60 ให้ใช้ AutoFit ตามปกติ
        End If
    Next rowNum
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน AdjustRowHeight_D11_D13 ใช้ในการปรับความสูงของแถว D11 ถึง D13 ให้อยู่ในขนาดเหมาะสม โดยกำหนดให้ความสูงอย่างน้อยต้องมี 60 point หากข้อความยาวเกินไประบบจะปรับอัตโนมัติ (AutoFit) เพื่อให้ข้อความในเซลล์แสดงครบถ้วน เหมาะสำหรับการจัดการรูปแบบข้อมูลให้ดูอ่านง่ายสม่ำเสมอ

2.8 Module12

2.8.1 ชื่อฟังก์ชัน SelectFolderAndSavePath

```
Sub SelectFolderAndSavePath()
    Dim FldrPicker As FileDialog
    Dim SelectedFolder As String
    Dim TargetCell As Range
    Set FldrPicker = Application.FileDialog(msoFileDialogFolderPicker) ' เรียกใช้งาน FileDialog แบบเลือกโฟลเดอร์
```

```
    With FldrPicker
        .Title = "กรุณาเลือกโฟลเดอร์สำหรับบันทึกภาพเรดาร์" ' ชื่อต่างหน้าต่าง FileDialog
        .AllowMultiSelect = False ' ไม่อนุญาตให้เลือกหลายโฟลเดอร์
        If .Show = -1 Then ' ถ้าผู้ใช้กดปุ่ม OK
            SelectedFolder = .SelectedItems(1) ' เก็บพาธของโฟลเดอร์ที่เลือกไว้
        Else
            Exit Sub ' ถ้าผู้ใช้กด Cancel ให้ยกเลิก
        End If
    End With
    Set TargetCell = ThisWorkbook.Sheets("Sheet1").Range("H11") ' กำหนดเซลล์เป้าหมายที่จะเก็บพาธ
    TargetCell.Value = SelectedFolder ' บันทึกพาธของโฟลเดอร์ลงในเซลล์
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน SelectFolderAndSavePath ใช้สำหรับเปิดหน้าต่างให้ผู้ใช้เลือกโฟลเดอร์จากเครื่องคอมพิวเตอร์ โดยเมื่อเลือกแล้วจะนำพาธของโฟลเดอร์นั้นไปเก็บไว้ในเซลล์ H11 ของชีต Sheet1 ซึ่งมักจะใช้ในกรณีที่ผู้ใช้ต้องการกำหนดตำแหน่งปลายทางสำหรับการบันทึกภาพเรดาร์หรือไฟล์อื่น ๆ

2.9 Module13

2.9.1 ชื่อฟังก์ชัน SaveRadarInfoAsJpg

```
Sub SaveRadarInfoAsJpg()
    Dim oWs As Worksheet
    Dim oRng As Range
    Dim oChrtO As ChartObject
```

```
    Dim lWidth As Long, lHeight As Long
    Dim savePath As String
    Dim picFileName1 As String, picFileName2 As String
    Set oWs = ActiveSheet ' ใช้แผ่นงานปัจจุบัน
    Set oRng = oWs.Range("B2:E16") ' กำหนดช่วงข้อมูลที่ต้องการจับภาพ
    savePath = oWs.Range("H11").Value ' อ่าน path ที่เก็บไว้ในเซลล์ H11
    If Dir(savePath, vbDirectory) = "" Then MsgBox "ไม่พบโฟลเดอร์ที่ระบุ!", vbExclamation,
"Error": Exit Sub ' ถ้าไม่พบโฟลเดอร์ให้แจ้งเตือน
    picFileName1 = savePath & "\RadarInfo.jpg" ' ชื่อไฟล์แบบคงที่
    picFileName2 = savePath & "\RadarInfo_" & Format(Now, "DD-MM-YYYY") & "_" &
oWs.Range("L5").Value & ".jpg" ' ชื่อไฟล์แบบมีวันที่และข้อมูลจาก L5
    oRng.CopyPicture xlScreen, xlPicture ' คัดลอกภาพช่วงเซลล์เป็นรูป
    lWidth = oRng.Width: lHeight = oRng.Height ' กำหนดความกว้างและสูงของกราฟให้พอดี
กับรูป
    Set oChrtO = oWs.ChartObjects.Add(Left:=0, Top:=0, Width:=lWidth,
Height:=lHeight) ' สร้าง ChartObject เปล่าสำหรับวางรูป
    oChrtO.Activate
    With oChrtO.Chart
        .Paste ' วางภาพลงในกราฟ
        .Export Filename:=picFileName1, Filtername:="JPG" ' บันทึกไฟล์ภาพแรก
        .Export Filename:=picFileName2, Filtername:="JPG" ' บันทึกไฟล์ภาพที่สอง
    End With
    oChrtO.Delete ' ลบ ChartObject ออกจากชีต
```

End Sub

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน SaveRadarInfoAsJpg ใช้บันทึกภาพของช่วงเซลล์ B2:E16 ซึ่งอาจเป็นข้อมูลรายงานเรดาร์ ให้เป็นไฟล์ .jpg โดยใช้เทคนิคการคัดลอกรูปภาพแล้ววางลงบนกราฟว่างเพื่อ export เป็นรูปภาพ จากนั้นจะบันทึกเป็นสองชื่อไฟล์ คือ RadarInfo.jpg แบบคงที่ และอีกไฟล์แบบมีวันที่และรหัสจากเซลล์ L5 เพื่อนำไปใช้งานต่อได้สะดวก ทั้งหมดนี้ขึ้นกับ path ที่กำหนดไว้ในเซลล์ H11 ว่ามีอยู่จริงหรือไม่

2.10 Module15

2.10.1 ชื่อฟังก์ชัน CopySavedImageToClipboard

```
Sub CopySavedImageToClipboard()
    Dim FilePath As String
    Dim savePath As String
    Dim ws As Worksheet
    Set ws = ActiveSheet ' ใช้แผ่นงานปัจจุบัน
    savePath = ws.Range("H11").Value ' อ่าน path จากเซลล์ H11
    FilePath = savePath & "\RadarInfo.jpg" ' กำหนด path ของไฟล์รูปที่ต้องการ copy
    If Dir(FilePath) = "" Then
        MsgBox "ไม่พบไฟล์รูปที่ต้องการ: " & FilePath, vbExclamation, "Error"
        Exit Sub ' ถ้าไม่พบไฟล์ให้หยุด
    End If
    Dim objShell As Object
    Set objShell = CreateObject("Shell.Application") ' เรียกใช้งาน Windows Shell
    objShell.Namespace(0).ParseName(FilePath).InvokeVerb ("Copy") ' สั่งให้คัดลอกรูปไปยัง Clipboard
```

```
    MsgBox "คัดลอกภาพ RadarInfo.jpg ไปยังคลิปบอร์ดแล้ว! ใช้ Ctrl+V เพื่อวางใน LINE หรือ
Facebook ได้เลย", vbInformation, "Success"

End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชัน CopySavedImageToClipboard ใช้คัดลอกไฟล์รูปภาพชื่อ RadarInfo.jpg ที่เก็บไว้ในโฟลเดอร์ซึ่งระบุในเซลล์ H11 ไปยัง Clipboard ของ Windows โดยใช้ Windows Shell API หลังจากคัดลอกแล้ว จะแสดงข้อความแจ้งเตือนผู้ใช้ว่าพร้อมให้วาง (Paste) ในแอปอื่น เช่น LINE หรือ Facebook ได้ทันที

2.11 Module6

2.11.1 ชื่อฟังก์ชัน ShapeCount_Click

```
Sub ShapeCount_Click()
    Dim shp As Shape ' ตัวแปรแทนรูปร่าง (Shape) ที่ถูกคลิก
    Dim ws As Worksheet ' ตัวแปรแทนแผ่นงาน
    Dim rowNum As Long ' ตัวแปรเก็บหมายเลขแถวที่ถูกคลิก
    Dim colLetter As String ' ตัวแปรเก็บชื่อคอลัมน์
    Dim lastNum As Long ' ตัวแปรเก็บค่าที่จะเพิ่มลงในเซลล์
    Set ws = ThisWorkbook.Sheets("Sheet1") ' ใช้ Sheet1 เป็นเป้าหมาย
    Set shp = ws.Shapes(Application.Caller) ' ระบุว่า Shape ไหนถูกคลิก
    rowNum = CLng(Mid(shp.Name, InStrRev(shp.Name, "N") + 1)) ' ดึงเลขแถวจากชื่อ
Shape เช่น C_N19
    If Sheet1.OptionButton1.Value = True Then
        colLetter = "N"
        lastNum = ws.Range("N18").Value + 1
```

```
            Rows("11").AutoFit
        ElseIf Sheet1.OptionButton2.Value = True Then
            colLetter = "O"
            lastNum = ws.Range("O18").Value + 1
            Rows("12").AutoFit
        ElseIf Sheet1.OptionButton3.Value = True Then
            colLetter = "P"
            lastNum = ws.Range("P18").Value + 1
            Rows("13").AutoFit
        Else
            MsgBox "กรุณาเลือกตัวเลือก", vbExclamation
            Exit Sub
        End If
        ws.Range(colLetter & rowNum).Value = lastNum ' บันทึกค่าเลขลงในเซลล์เป้าหมาย
    End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชันนี้ใช้ใส่หมายเลขลงในคอลัมน์ N, O หรือ P ตามลำดับที่เลือก โดยดึงหมายเลขจากแถว 18 และนำไปใส่ในเซลล์ของแถวที่คลิก Shape

2.11.2 ชื่อฟังก์ชัน ShapeDel_Click

```
Sub ShapeDel_Click()
    Dim shp As Shape ' ตัวแปรแทนรูปร่าง (Shape) ที่ถูกคลิก
    Dim ws As Worksheet ' ตัวแปรแทนแผ่นงาน
    Dim rowNum As Long ' ตัวแปรเก็บหมายเลขแถวที่ถูกคลิก
```

```
    Dim colLetter As String ' ตัวแปรเก็บชื่อคอลัมน์
    Dim lastNum As Long ' ตัวแปรเก็บค่าล่าสุด (ไม่ได้ใช้ในคำสั่งลบ แต่ยังมีการกำหนดไว้)
    Set ws = ThisWorkbook.Sheets("Sheet1") ' ใช้ Sheet1 เป็นเป้าหมาย
    Set shp = ws.Shapes(Application.Caller) ' ระบุว่า Shape ไหนถูกคลิก
    rowNum = CLng(Mid(shp.Name, InStrRev(shp.Name, "N") + 1)) ' ดึงเลขแถวจากชื่อ Shape เช่น C_N19
    If Sheet1.OptionButton1.Value = True Then
        colLetter = "N"
        lastNum = ws.Range("N18").Value + 1
    ElseIf Sheet1.OptionButton2.Value = True Then
        colLetter = "O"
        lastNum = ws.Range("O18").Value + 1
    ElseIf Sheet1.OptionButton3.Value = True Then
        colLetter = "P"
        lastNum = ws.Range("P18").Value + 1
    Else
        MsgBox "กรุณาเลือกตัวเลือก", vbExclamation
        Exit Sub
    End If
    ws.Range(colLetter & rowNum).ClearContents ' ลบค่าที่อยู่ในเซลล์ที่กำหนด
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชันนี้ใช้สำหรับลบค่าที่เคยบันทึกไว้ในเซลล์คอลัมน์ N, O หรือ P โดยดูจากแถวของ Shape ที่ถูกคลิกและตัวเลือกที่เลือก

2.11.3 ชื่อฟังก์ชัน ClearColor

Sub ClearColor()

If Not shp Is Nothing Then shp.Fill.Transparency = 1: shp.Line.ForeColor.RGB = RGB(0, 0, 0): Call Module6.ShapeDel_Click

End If

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชันนี้จะทำให้ Shape ที่เคยใส่สีอยู่กลายเป็นโปร่งใส และเรียกใช้ ShapeDel_Click เพื่อลบค่าที่เกี่ยวข้องในตาราง

2.11.4 ชื่อฟังก์ชัน Draw_Thick_Line_Arrow

```
Public Sub Draw_Thick_Line_Arrow()
    Static IsStartPoint As Boolean ' ใช้เก็บสถานะว่ากำลังเลือกจุดเริ่มต้นหรือจุดสิ้นสุด
    Static StartX As Double, StartY As Double ' พิกัดจุดเริ่มต้น
    Dim CursorPos As POINTAPI ' พิกัดเมาส์บนหน้าจอ (pixel)
    Dim ExcelPos As POINTAPI ' พิกัดมุมบนซ้ายของหน้าต่าง Excel (pixel)
    Dim EndX As Double, EndY As Double ' พิกัดจุดสิ้นสุด (point)
    Dim hdc As LongPtr ' Handle ของหน้าจอ ใช้คำนวณ DPI
    Dim PointsPerPixelX As Double, PointsPerPixelY As Double ' สัดส่วนการแปลง pixel
→ point
    Dim ws As Worksheet
    Dim ArrowName As String ' ชื่อลูกศร
```

```
    Dim StartMarker As Shape ' วัตถุแสดงตำแหน่งจุดเริ่มต้น
    Static ArrowCount As Integer ' ตัวนับจำนวนลูกศร
    Set ws = ThisWorkbook.Sheets("Sheet1")
    ws.Unprotect Password:="" ' ปลดล็อกชีตก่อนวาด
    hdc = GetDC(0)
    If hdc <> 0 Then
        PointsPerPixelX = 72 / GetDeviceCaps(hdc, LOGPIXELSX) ' คำนวณ DPI แนวนอน
        PointsPerPixelY = 72 / GetDeviceCaps(hdc, LOGPIXELSY) ' คำนวณ DPI แนวตั้ง
        ReleaseDC 0, hdc
    Else
        MsgBox "ไม่สามารถดึงข้อมูลหน้าจอได้", vbCritical, "Error"
        Exit Sub
    End If
    PointsPerPixelX = PointsPerPixelX * 100 / ActiveWindow.Zoom ' ปรับค่าตาม Zoom
แนวนอน
    PointsPerPixelY = PointsPerPixelY * 100 / ActiveWindow.Zoom ' ปรับค่าตาม Zoom
แนวตั้ง
    ExcelPos.X = ActiveWindow.PointsToScreenPixelsX(0) ' หาพิกัดซ้ายบนของ Excel บน
หน้าจอ
    ExcelPos.Y = ActiveWindow.PointsToScreenPixelsY(0)
    If GetCursorPos(CursorPos) = 0 Then Exit Sub ' รับตำแหน่งเมาส์ (pixel) จากหน้าจอ
    If Not IsStartPoint Then
```

```
        StartX = (CursorPos.X - ExcelPos.X) * PointsPerPixelX ' แปลงพิกัด X จาก pixel → point
        StartY = (CursorPos.Y - ExcelPos.Y) * PointsPerPixelY
        Set StartMarker = ws.Shapes.AddShape(msoShapeOval, StartX - 3, StartY - 3, 6, 6) ' วาดจุดเริ่ม (วงกลมแดง)
        With StartMarker
            .Fill.ForeColor.RGB = RGB(255, 0, 0)
            .Line.Visible = msoFalse
            .Name = "TempMarker"
        End With
    Else
        EndX = (CursorPos.X - ExcelPos.X) * PointsPerPixelX
        EndY = (CursorPos.Y - ExcelPos.Y) * PointsPerPixelY
        ArrowCount = ArrowCount + 1
        ArrowName = "Arrow" & ArrowCount ' สร้างชื่อลูกศรใหม่
        Dim newArrow As Shape
        Set newArrow = ws.Shapes.AddConnector(msoConnectorStraight, StartX, StartY, EndX, EndY) ' วาดเส้นตรง
        With newArrow.Line
            .ForeColor.RGB = RGB(255, 255, 255) ' สีขาว
            .Weight = 10 ' ความหนา 10pt
            .EndArrowheadStyle = msoArrowheadTriangle ' หัวลูกศรแบบสามเหลี่ยม
            .EndArrowheadWidth = 2 ' กว้าง
```

```
            .EndArrowheadLength = 1 ' ยาว
            .Transparency = 0.5 ' โปร่งแสง 50%
        End With
        With newArrow.Glow
            .Color.ObjectThemeColor = msoThemeColorAccent2 ' สี Glow ตาม Theme
    Accent2
            .Radius = 3
            .Transparency = 0.5
        End With
        newArrow.Name = ArrowName
        newArrow.OnAction = "DeleteSelf" ' เมื่อคลิกจะเรียกฟังก์ชัน DeleteSelf
        On Error Resume Next
        ws.Shapes("TempMarker").Delete ' ลบวงกลมจุดเริ่ม
        On Error GoTo 0
    End If
    IsStartPoint = Not IsStartPoint ' เปลี่ยนสถานะสำหรับคลิกถัดไป
    ws.Protect Password:="" ' ล็อกชีตกลับ
End Sub
```

คำอธิบายหน้าที่ของฟังก์ชัน

ฟังก์ชันนี้ใช้วาดเส้นลูกศรลงในชีต Excel ด้วยการคลิกเมาส์ 2 ครั้ง: ครั้งแรกเลือกจุดเริ่มต้น (มีวงกลมแดงแสดงไว้) และครั้งที่สองกำหนดจุดปลายเพื่อสร้างลูกศรหนาพร้อมเอฟเฟกต์ Glow โดยใช้ API อ่านตำแหน่งเมาส์จากหน้าจอ ⟶ แปลงเป็นหน่วย point ⟶ วาดเส้น ⟶ ตั้งชื่อ ⟶ ผูก OnAction ให้คลิกเพื่อลบได้ในภายหลัง

3. User Defined Function (UDF)

3.1 ฟังก์ชัน GetProvinceSummary

```
Function GetProvinceSummary(dataRange As Range, orderRange As Range) As String
Dim dict As Object, provinceOrder As Object, districtList As Object
Set dict = CreateObject("Scripting.Dictionary")
Set provinceOrder = CreateObject("Scripting.Dictionary")
Set districtList = CreateObject("Scripting.Dictionary")
Dim i As Integer, province As String, district As String, order As Variant
Dim result As String, sortedProvinces As Variant, sortedDistricts As Variant
If dataRange.Rows.Count <> orderRange.Rows.Count Then GetProvinceSummary = "ERROR: ช่วงข้อมูลไม่สอดคล้องกัน": Exit Function
For i = 1 To dataRange.Rows.Count
    order = orderRange.Cells(i, 1).Value ' ลำดับที่ใช้สำหรับเรียง
    district = Trim(dataRange.Cells(i, 1).Value) ' ชื่ออำเภอ
    province = Trim(dataRange.Cells(i, 2).Value) ' ชื่อจังหวัด
    If IsEmpty(order) Or province = "" Then GoTo NextIteration
    If Not provinceOrder.exists(province) Then provinceOrder.Add province, order Else If order < provinceOrder(province) Then provinceOrder(province) = order
    Dim tempArray As Variant
    If districtList.exists(province) Then tempArray = districtList(province) Else tempArray = Array()
```

```
    If Not IsArray(tempArray) Or UBound(tempArray) < 0 Then tempArray =
Array(Array(order, district)) Else ReDim Preserve tempArray(UBound(tempArray) + 1):
tempArray(UBound(tempArray)) = Array(order, district)
    districtList(province) = tempArray
NextIteration:
Next i
If provinceOrder.Count > 0 Then
    sortedProvinces = provinceOrder.keys
    QuickSort sortedProvinces, LBound(sortedProvinces), UBound(sortedProvinces),
provinceOrder
Else
    GetProvinceSummary = "-"
    Exit Function
End If
Dim key As Variant
For Each key In sortedProvinces
    sortedDistricts = districtList(key)
    If Not IsArray(sortedDistricts) Or UBound(sortedDistricts) < 0 Then GoTo
NextProvince
    QuickSortDistricts sortedDistricts, LBound(sortedDistricts), UBound(sortedDistricts)
    Dim districts As String
    districts = ""
    For i = LBound(sortedDistricts) To UBound(sortedDistricts)
```

```
        districts = districts & " " & sortedDistricts(i)(1)
    Next i
    If Trim(districts) <> "" Then result = result & " <<" & key & ">> " & Trim(districts) &
vbNewLine
NextProvince:
Next key
GetProvinceSummary = Trim(result)
End Function
```

3.1.1 ฟังก์ชัน QuickSort สำหรับเรียงชื่อจังหวัด

```
Sub QuickSort(arr As Variant, first As Long, last As Long, dict As Object)
Dim pivot As Variant, temp As Variant
Dim i As Long, j As Long
If first >= last Then Exit Sub
pivot = dict(arr((first + last) \ 2)) ' ใช้ค่าลำดับจาก dictionary เป็น pivot
i = first: j = last
Do
    Do While dict(arr(i)) < pivot: i = i + 1: Loop
    Do While dict(arr(j)) > pivot: j = j - 1: Loop
    If i <= j Then temp = arr(i): arr(i) = arr(j): arr(j) = temp: i = i + 1: j = j - 1
Loop While i <= j
If first < j Then QuickSort arr, first, j, dict
If i < last Then QuickSort arr, i, last, dict
End Sub
```

3.1.2 ฟังก์ชัน QuickSortDistricts สำหรับเรียงอำเภอในแต่ละจังหวัด

```
Sub QuickSortDistricts(arr As Variant, first As Long, last As Long)
Dim pivot As Variant, temp As Variant
Dim i As Long, j As Long
If first >= last Then Exit Sub
pivot = arr((first + last) \ 2)(0) ' ใช้ค่าลำดับใน array ของอำเภอเป็น pivot
i = first: j = last
Do
    Do While arr(i)(0) < pivot: i = i + 1: Loop
    Do While arr(j)(0) > pivot: j = j - 1: Loop
    If i <= j Then temp = arr(i): arr(i) = arr(j): arr(j) = temp: i = i + 1: j = j - 1
Loop While i <= j
If first < j Then QuickSortDistricts arr, first, j
If i < last Then QuickSortDistricts arr, i, last
End Sub
```

คำอธิบายการทำงาน

GetProvinceSummary เป็นฟังก์ชันแบบ User Defined Function (UDF) ที่สามารถนำมาใช้กับเซลล์ในเวิร์กชีต Excel ได้ โดยรับข้อมูลเป็นสองช่วง (Range):

- **orderRange**: ช่วงที่ระบุลำดับการคลิก เช่น ค่าที่ได้จากการคลิก Shape
- **dataRange**: ช่วงข้อมูลที่ประกอบด้วยชื่ออำเภอ (คอลัมน์ 1) และจังหวัด (คอลัมน์ 2)

ภายในฟังก์ชันจะใช้:

- **provinceOrder** เพื่อบันทึกลำดับที่น้อยที่สุดของแต่ละจังหวัด ใช้สำหรับเรียงลำดับจังหวัด
- **districtList** เพื่อเก็บรายการอำเภอของแต่ละจังหวัด พร้อมลำดับการคลิกของแต่ละอำเภอ

อัลกอริธึมการจัดเรียง:

- เรียงจังหวัดโดยใช้ QuickSort โดยอิงตามลำดับที่น้อยที่สุดที่พบในแต่ละจังหวัด
- เรียงอำเภอภายในแต่ละจังหวัดโดยใช้ QuickSortDistricts ตามลำดับที่คลิก

ผลลัพธ์คือข้อความ (String) ที่สรุปรายชื่ออำเภอที่คลิก โดยเรียงตามจังหวัดและลำดับที่คลิก เช่น:

<<ขอนแก่น>> เมืองขอนแก่น บ้านไผ่

<<นครราชสีมา>> เมือง โนนสูง

เมื่อนำไปใช้ในเซลล์ เช่น

=GetProvinceSummary(A1:A10,B1:B10)
จะได้ผลสรุปที่จัดเรียงใหม่ตามลำดับที่คลิกและกลุ่มจังหวัดอย่างถูกต้อง

ฟังก์ชันนี้อยู่ใน **Module5** และเป็นหัวใจหลักของการสร้างรายงานสรุป "อำเภอที่มีฝนตก" ในโปรแกรม**ClickRain** โดยไม่ต้องเขียนโค้ดเพิ่มทุกครั้งที่อัปเดตการคลิก