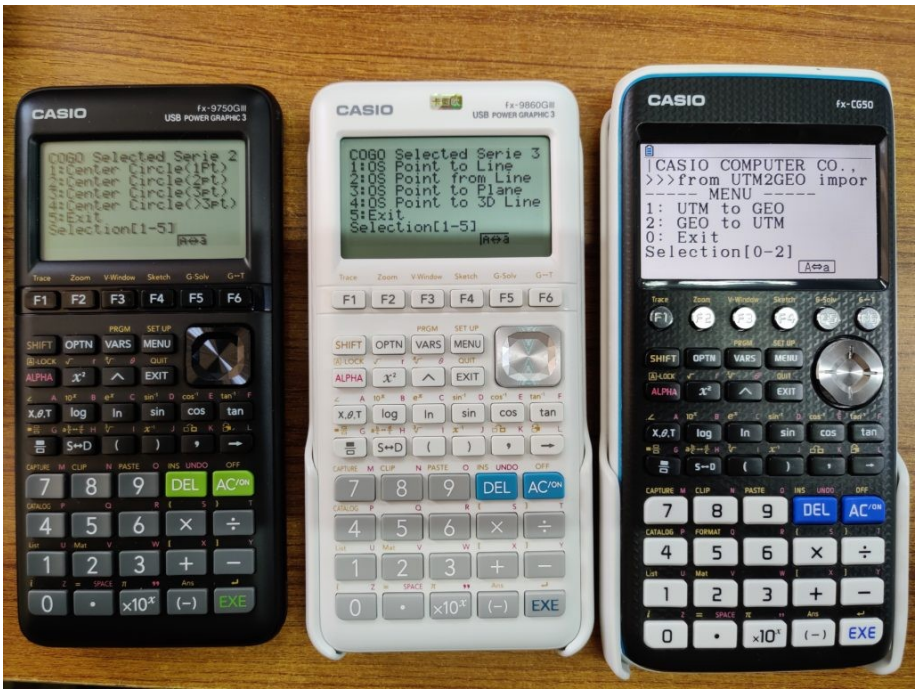
โปรแกรมภาษาไพทอนบนเครื่องคิดเลขคาสิโอ fx-9750GIII fx-9860GIII และ fx-cg50 โปรแกรมพื้นฐานงานสำรวจชุดที่ 1 (COGO Selected Serie 1)

ไม่กี่วันนี้ผมได้ถอยเครื่องคิดเลข fx-9860GIII มาหนึ่งเครื่องราคาประมาณสี่พันห้าร้อยบาท ส่วนน้องๆในที่ทำงานถอย fx-9750GIII มาหนึ่งเครื่องเช่นเดียวกันแต่ราคาย่อมเยากว่า ราคาเครื่องประมาณสามพันบาท สองรุ่นนี้เขียนภาษาไพทอนได้ ไพทอนที่ลงในเครื่องคิดเลขเป็นไพทอนรุ่นเล็กเรียกว่า ไมโครไพทอน (Micropython) แต่ไมโครไพทอนที่ลงในเครื่องคิดเลข ทางคาสิโอลงไลบรารีมาให้ใช้แค่สองไลบรารีคือ math และ random ที่อยากได้มากคือไลบรารี io ที่สามารถเขียนอ่านไฟล์ได้กลับไม่ลงมาให้ ทำให้การใช้งานจำกัดจำเขี่ยเหมือนโดนมัดมือมัดเท้า เอาละไม่เป็นไร ผมจะลองเขียนโปรแกรมไพทอนและพยายามใช้ทรัพยากรอันจำกัดจำเขี่ยนี้ให้ออกมาดูดีที่สุด นี่เป็นความท้าทายเล็กๆน้อยๆอย่างหนึ่ง

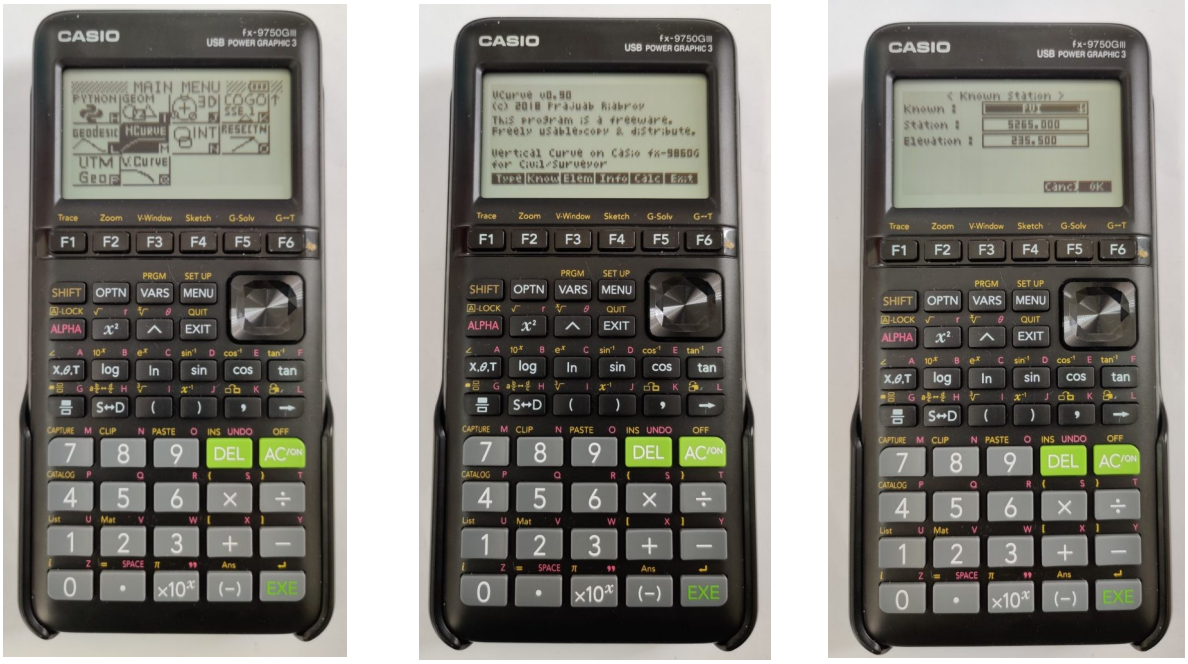
ณ ตอนนี้มีเครื่องคิดเลขคาสิโอสามรุ่นที่สนับสนุนภาษาไพทอนเรียงกันดังนี้ fx-9750GIII, fx-9860GIII และ fx-cg50



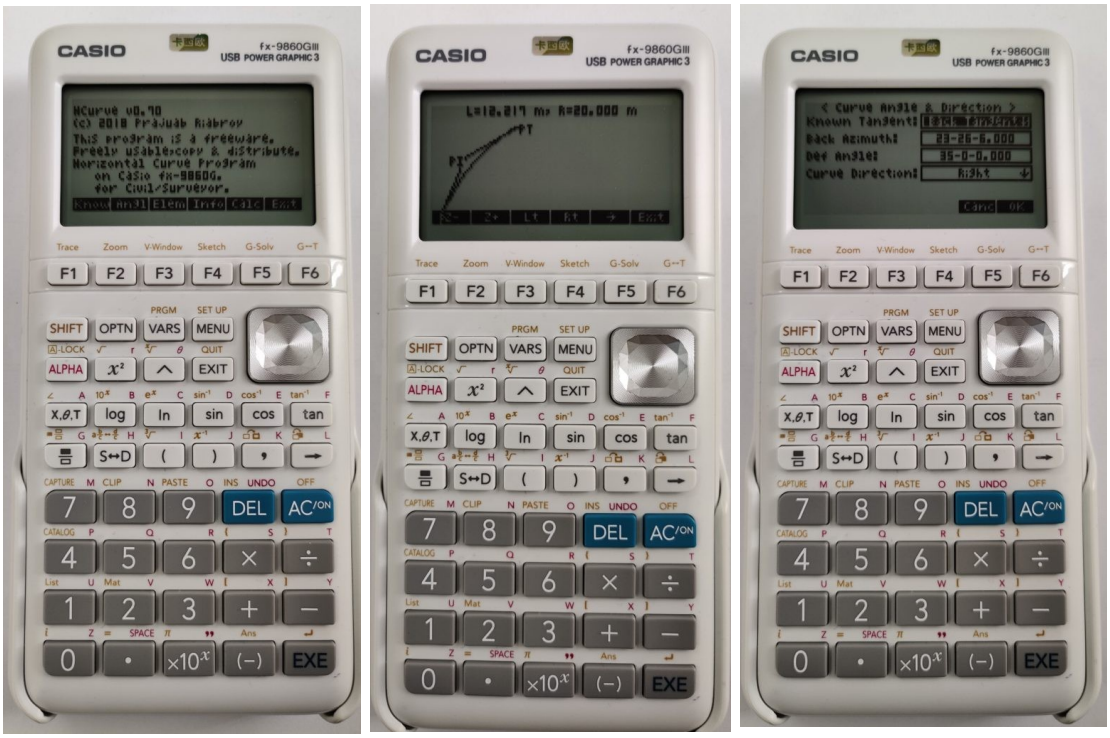
เครื่องคิดเลขคาสิโอ fx-9750GIII, fx-9860GIII และ fx-cg50

ทดสอบโปรแกรมงานสำรวจที่พัฒนาด้วยภาษาซีบน fx-9750GIII และ fx-9860GIII

ขอย้อนไปโปรแกรมงานสำรวจพื้นฐานบนเครื่องคิดเลข **fx-9860GII** ที่ผมเขียนด้วยภาษาซี ผมลองเอามาลงในเครื่อง **fx-9750III** และ **fx-9860GIII** รันได้ปกติแสดงว่าสถาปัตยกรรมเดียวกัน fx-9860GII ใช้ซีพียู SH4a ส่วนรุ่นอื่นไม่ชัดเจน ความต่างกันที่แน่ๆคือหน่วยความจำ



โปรแกรมงานสำรวจพัฒนาด้วยภาษาซี ใช้ได้บนคาสิโอ fx-9750GIII



โปรแกรมงานสำรวจพัฒนาด้วยภาษาซี ใช้ได้บนคาสิโอ fx-9860GIII

ชุดโปรแกรมพื้นฐานงานสำรวจ COGO (Coordinate Geometry) ชุดที่ 1

โปรแกรมที่จะเขียนจัดอยู่ในชุด **COGO (Coordinate Geometry)** ที่มาแก้ปัญหาพิกัดเรขาคณิตในงานสำรวจนั่นเอง ผมเคยเขียนโปรแกรมชุดนี้ด้วยภาษาซีบนเครื่องคิดเลข fx-9860GII (SD) รุ่นก่อน ลองมาดูในภาคภาษาไพทอนบ้าง ผมจะลงซอร์สโค้ดให้ท้ายๆบทความ เผื่อผู้อ่านมีเครื่องคิดเลขรุ่นเหล่านี้จะสามารถก็อปปี้ไปลงเครื่องคิดเลขตัวเองได้

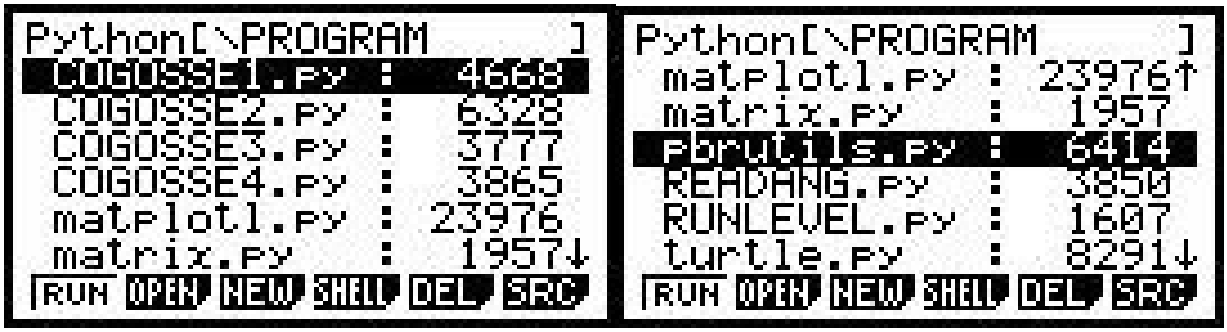
ส่วนประกอบของโปรแกรม

สำหรับโปรแกรมพื้นฐานงานสำรวจในชุดนี้จะจัดโปรแกรมย่อยเล็กๆ ไว้ 4 โปรแกรม

1. **Bearing-Dist (2 pt)** เมื่อกำหนดจุดค่าพิกัด 2 จุด สำหรับคำนวณหามุมอะซิมัทและระยะทาง
2. **Bearing-Dist(3 pt) เ**มื่อกำหนดจุดค่าพิกัด 3 จุด สำหรับคำนวณหาง่ามมุมราบ อะซิมัทและระยะทาง ในงานสำรวจก็ได้แก่การตั้งเป้าหลัง (backsight) จุดตั้งกล้อง (station) และเป้าหน้า (target)
3. **Coordinate 2D** เมื่อกำหนดจุดค่าพิกัด 2 จุด กำหนดมุมราบและระยะราบ คำนวณหาค่าพิกัดจุดที่ 3 คำนวณหาพิกัดจุดที่ 3 การคำนวณคำนวณในระนาบสองมิติอย่างเดียว ไม่มีค่าระดับมาเกี่ยวข้อง
4. **Coordinate 3D** เมื่อกำหนดจุดค่าพิกัด 2 จุด กำหนดมุมราบและมุมดิ่ง ระยะทางแบบ slope distance ต้องการคำนวณหาค่าพิกัดและค่าระดับจุดที่ 3

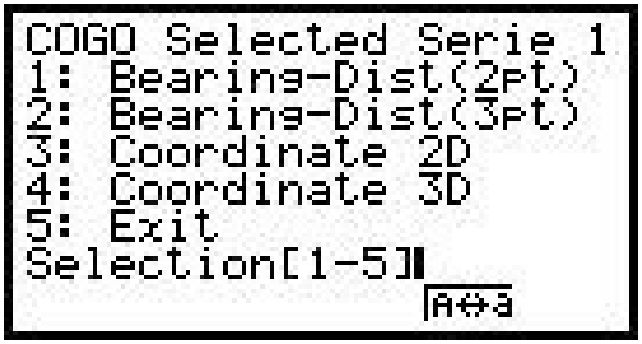
วิธีการใช้งานโปรแกรม

กดคีย์ “**MENU**” ที่เครื่องคิดเลข กดคีย์ลูกศรเลื่อนไปที่ไอคอน “Python” กดเข้าไป ตรงนี้ผมสร้างโฟลเดอร์ชื่อ “PROGRAM” ผมเอาซอร์สโค้ดโปรแกรมไพทอนมาไว้ที่นี่ เมื่อกด “EXE” เข้าไปจะเห็นไฟล์ซอร์สโค้ดไพทอนเรียงกันประมาณนี้ ตอนนี้จะรันไฟล์ COGOSSE1.py ซึ่งจะต้องมีไฟล์ไลบรารี pbrutils.py ด้วย



โปรแกรมพื้นฐานงานสำรวจชุดที่ 1 (COGO Selected Serie 1)

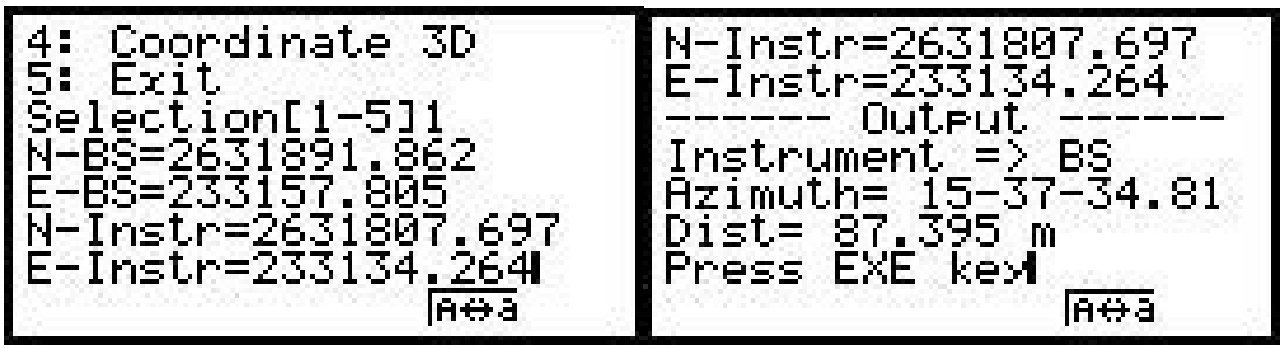
เลื่อนแถบเคอร์เซอร์ไปที่ COGOSSE1.py กดคีย์ “F1” หรือกดคีย์ “EXE” จะเห็นเมนู



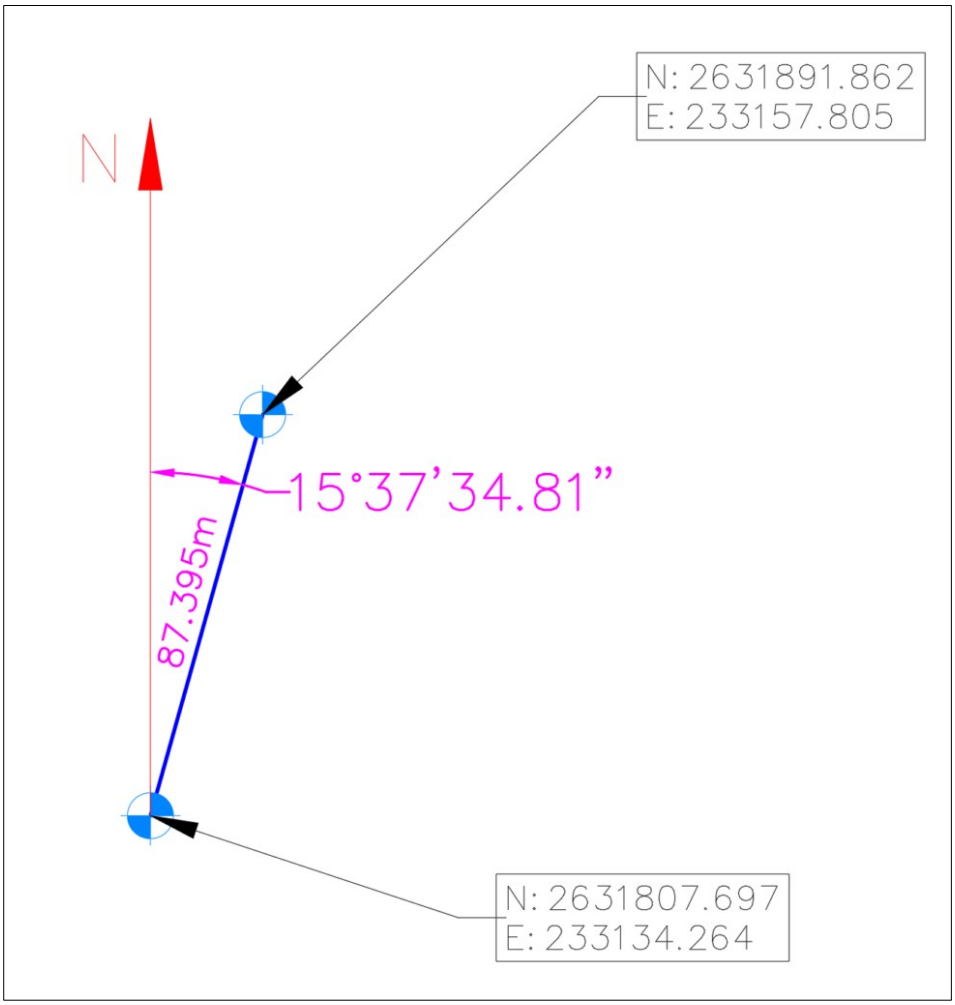
เมนูโปรแกรม พื้นฐานงานสำรวจชุดที่ 1 (COGO Selected Serie 1)

Bearing-Dist (2 pt)

ที่เมนูกดคีย์ “1” และกดคีย์ “EXE” เป็นการคำนวณหา**ค่ามุมอะซิมัทและระยะทาง**เมื่อกำหนดจุดค่าพิกัดให้สองจุด ลองทดสอบจากตัวอย่างดังรูป การประยุกต์ใช้งานส่วนใหญ่เป็นตอนที่ช่างสำรวจตั้งกล้องที่หมุดและส่องไปเป้าหลังหรือเป้าหน้าแล้ววัดระยะทางเพื่อตรวจสอบค่าพิกัดเทียบกับค่าที่คำนวณด้วยเครื่องคิดเลข ทดสอบป้อนค่าและจะได้ผลลัพธ์ออกมา จะได้มุมอะซิมัทจากจุดตั้งกล้องไปเป้าหลัง 15°3′ 34.81″ *(ตั้งทิศเหนือที่จุดตั้งต้น ที่นี่หมายถึงจุดตั้งกล้องกวาดมุมตามเข็มนาฬิกาไปหาแขนต้นทาง-ปลายทาง)* ระยะทาง 87.395 เมตร กดคีย์ “EXE” อีกครั้งโปรแกรมจะออกมาที่เมนูเดิม



ป้อนค่าและผลลัพธ์ของ Bearing – Dist (2 pt)

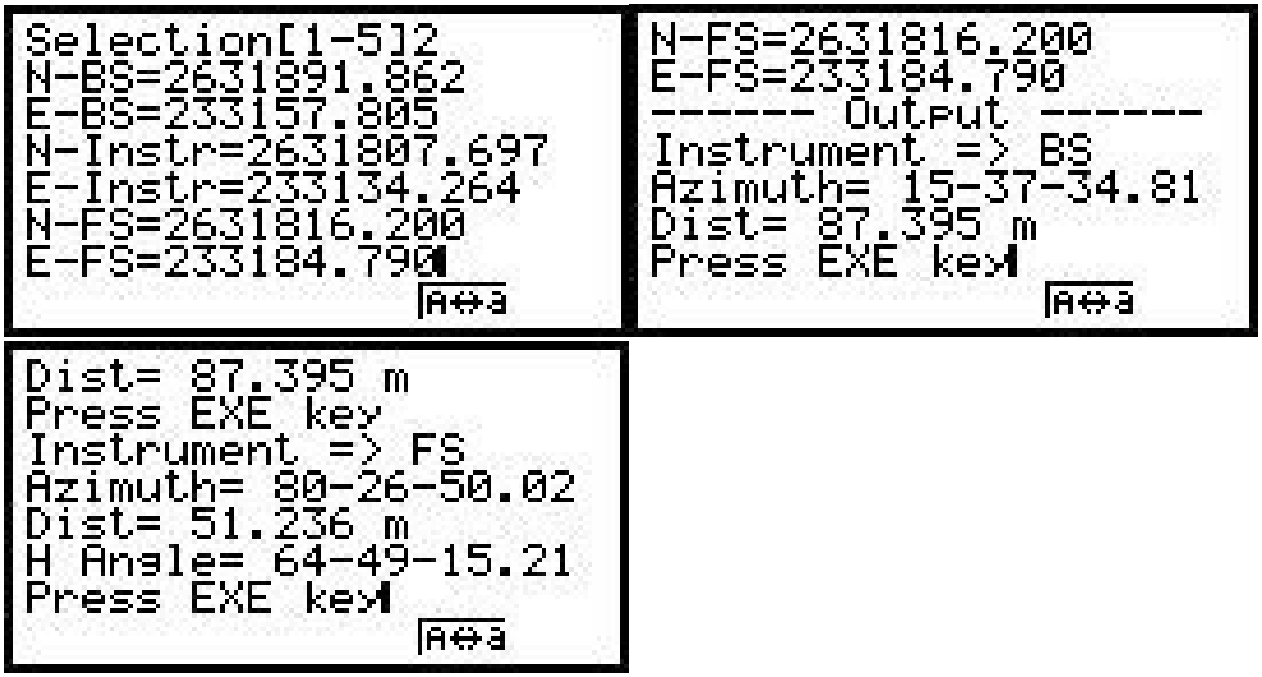


แผนผังแสดงตัวอย่างของ Bearing – Dist (2 pt)

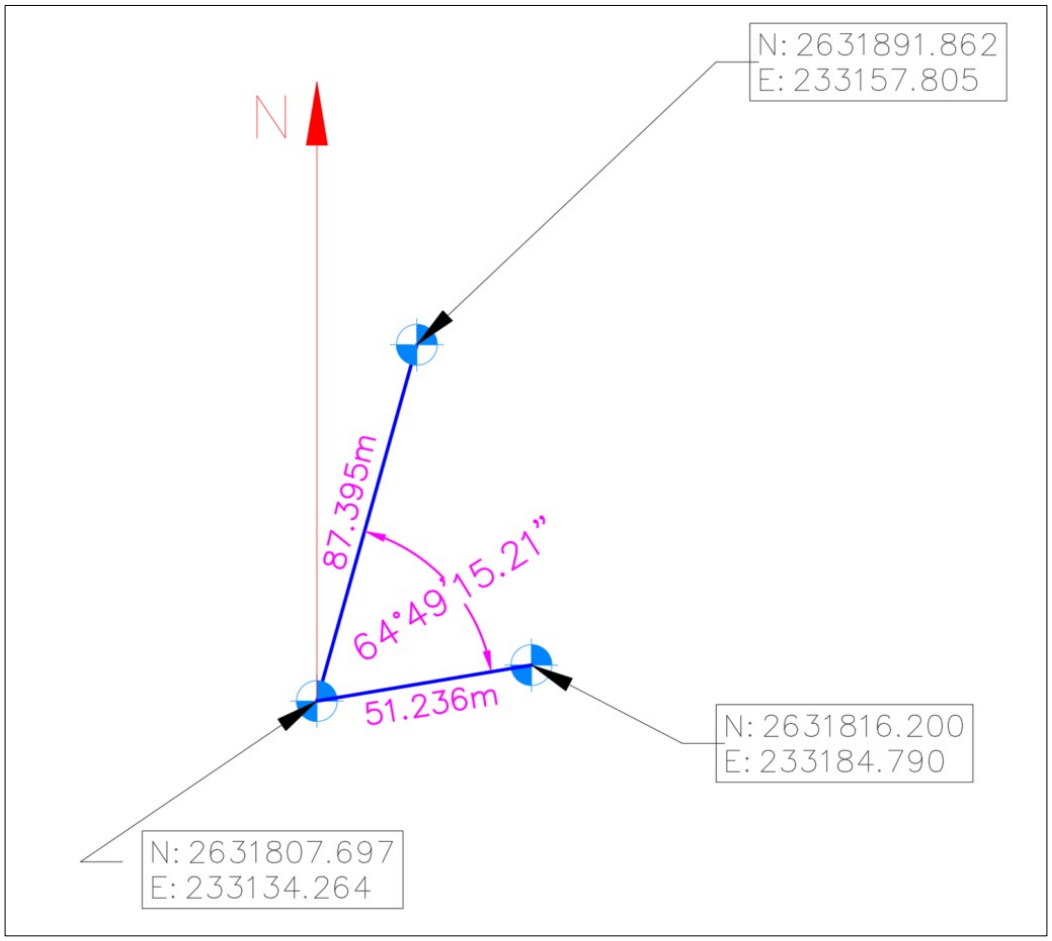
Bearing-Dist (3 pt)

ที่เมนูกดคีย์เลข “2” และกดคีย์ “EXE” การประยุกต์ใช้งานส่วนใหญ่จะเป็นการตั้งกล้องส่องไปหมุดเป้าหลังแล้วป้อนค่าพิกัดเป้าหน้าเพื่อตรวจสอบมุมราบหรือไม่ก็จะเป็นการวางผังโดยการเปิดมุมราบและวัดระยะทางที่เป้าบน pole

ลองดูตัวอย่างทดสอบ จะได้ผลลัพธ์มาดังนี้ ครั้งแรกจะแสดงมุมอะซิมัทและระยะทางไปเป้าหลังก่อน ถัดไปจะเป็นมุมราบ มุมอะซิมัทและระยะทางไปเป้าหน้า



ป้อนค่าและผลลัพธ์ของ Bearing – Dist (3 pt)

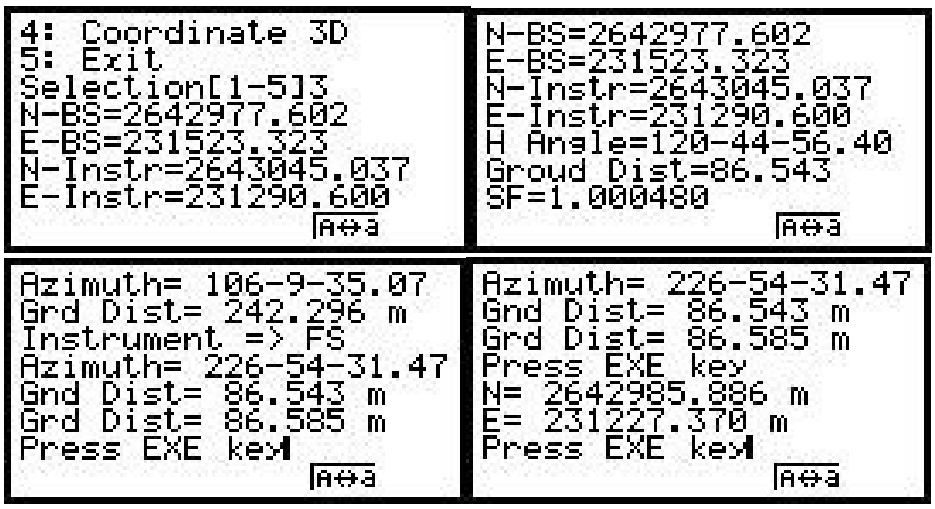


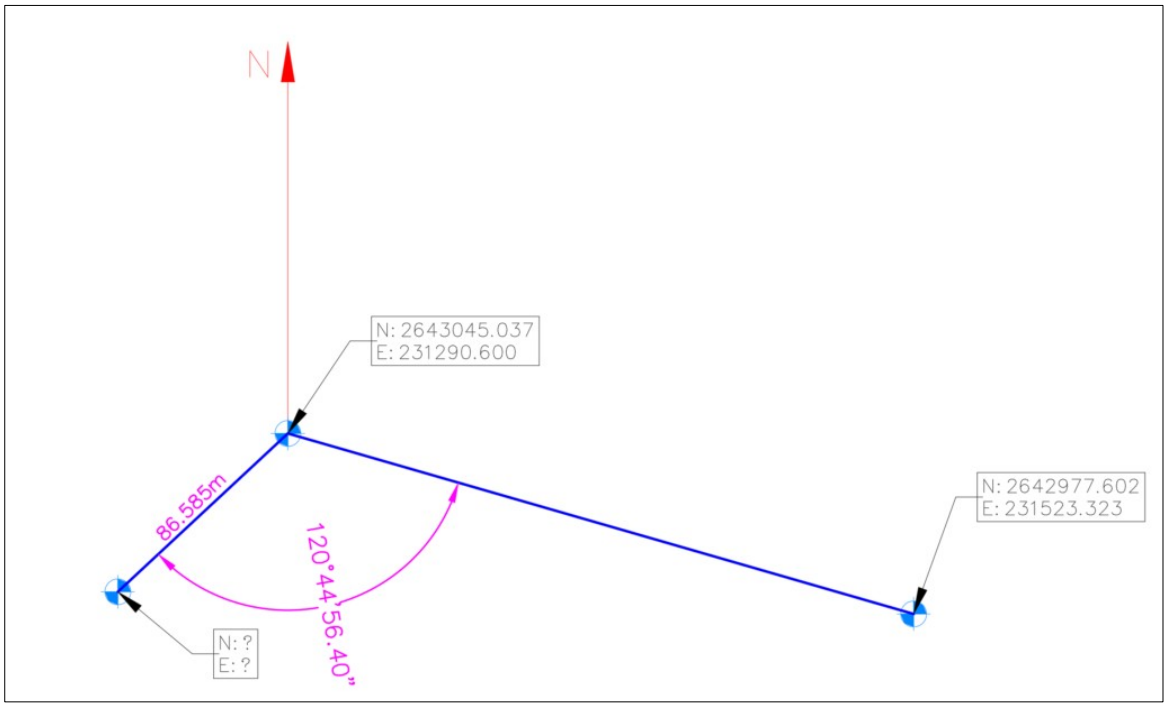
แผนผังแสดงตัวอย่าง Bearing – Dist (3 pt)

Coordinate 2D

ที่เมนูกดคีย์เลข “3” และกดคีย์ “EXE” เป็นการคำนวณหาค่าพิกัดเป้าหน้าเมื่อกำหนดค่าพิกัดจุดตั้งกล้องและเป้าหลัง กำหนดมุมราบและระยะทาง การคำนวณจะไม่มีค่าระดับมาเกี่ยวข้อง จึงเรียกว่า 2D หรือสองมิติ สำหรับโปรแกรมนี้ผมได้นำค่าสเกลแฟคเตอร์เข้ามาช่วยประยุกต์ใช้ด้วย ในกรณีที่ไม่ต้องใช้ก็ป้อนค่าสเกลแฟคเตอร์นี้ เป็น 1.0

สเกลแฟคเตอร์ตัวนี้แล้วจริงๆคือ **Combine Scale Factor (CSF)** ที่ได้จาก **Elevation Scale Factor (ESF) x Grid Scale Factor (GSF)** การประยุกต์ใช้สเกลแฟคเตอร์ส่วนใหญ่นำมาใช้โครงการที่ระบบพิกัดฉากกริดยูทีเอ็มในงานใหญ่ๆยาวๆ เช่นโครงการก่อสร้างถนน รถไฟ เพราะว่าแบบ drawing เราอยู่บนระนาบพิกัดฉาก ให้คิดเสียว่าแบบอยู่บนกระดาษขนาดใหญ่มาตราส่วน 1:1 แล้วเราวัดระยะทางบนผิวโลกที่มีความโค้ง ดังนั้นการวัดระยะทางจะต้องมีการทอนจากบนผิวโค้งเพื่อให้ลงมาเข้ากับระนาบพิกัดฉากของกระดาษ มาลองทดสอบข้อมูล ป้อนข้อมูลค่าพิกัดเป้าหลัง ค่าพิกัดจุดต้องกลอง จากนั้นป้อนมุมป้อนระยะทางและสเกลแฟคเตอร์ (ถ้าใช้) แล้วจะได้ผลลัพธ์เป็นค่าพิกัด **N=2642985.886, E=231227.370**

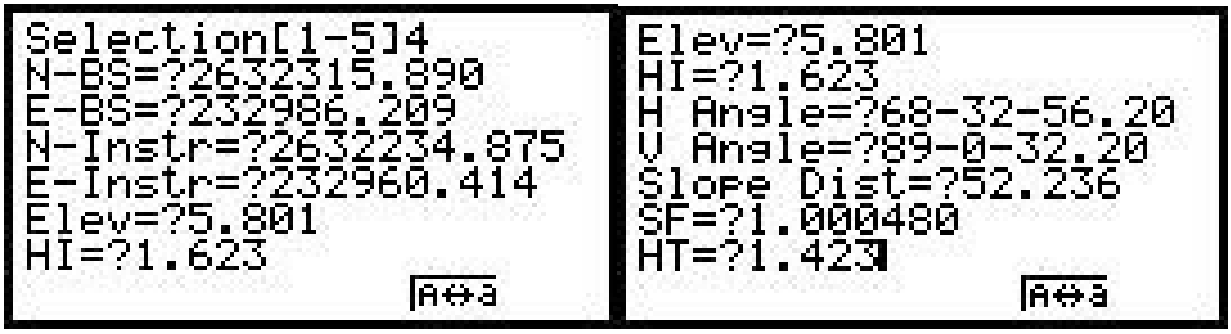




ป้อนค่าและผลลัพธ์ Coordinate 2D แผนผังแสดงตัวอย่าง Coordinate 2D

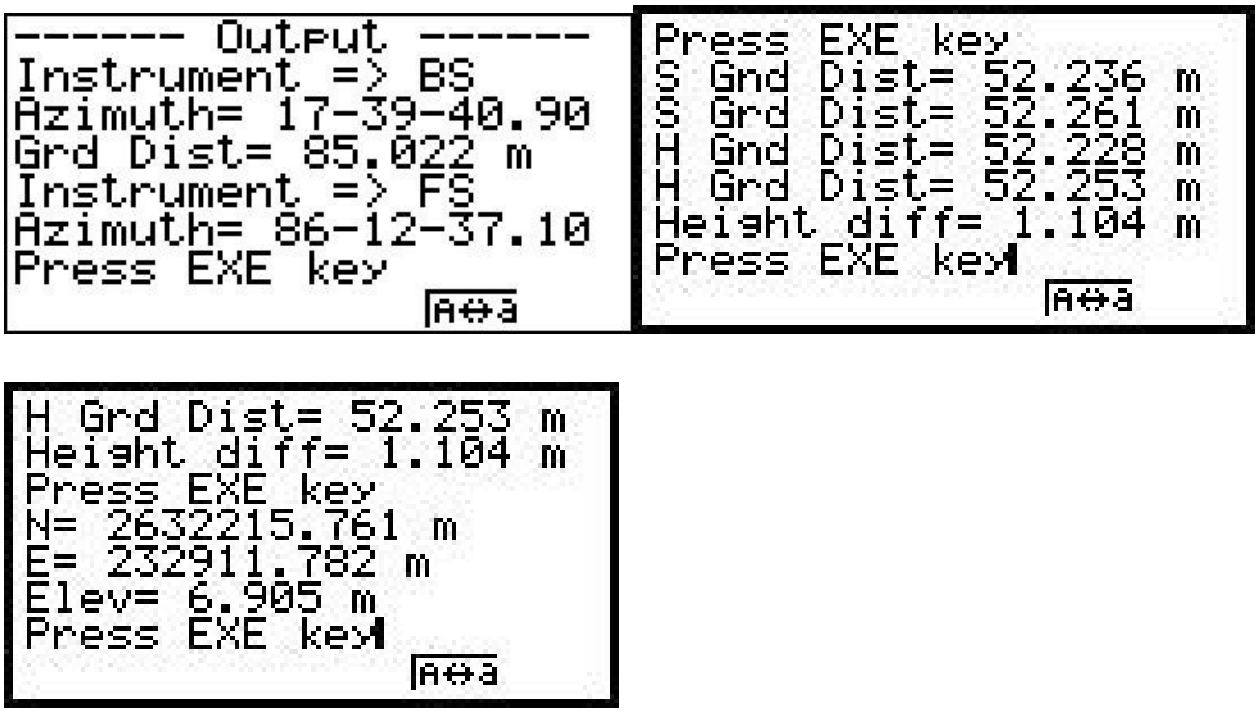
Coordinate 3D

ที่เมนูกดคีย์ “4” และกดคีย์ “EXE” โปรแกรมคล้าย Coordinate 2D แต่จะมีมิติทางดิ่งเข้ามาเพิ่มดังนั้นที่จุดตั้งกล้องจะวัดความสูงของกล้อง (HI – Height of instrument) และที่เป้าหน้าก็จะต้องวัดความสูงมาด้วย (HT – Height of target) นอกจากนั้นจะมีมุมดิ่ง (Vertical angle) มาด้วย มาดูข้อมูลทดสอบกัน เริ่มจากป้อนค่าพิกัดเป้าหลัง ต่อมาป้อนค่าพิกัดจุดตั้งกล้อง ค่าระดับจุดตั้งกล้อง ความสูงกล้อง ต่อไปป้อนมุมราบ(H.Ang) มุมดิ่ง(V.Ang) ระยะทาง (Slope distance) และความสูงเป้า(HT) และค่าสเกลแฟคเตอร์ (Scale Factor)

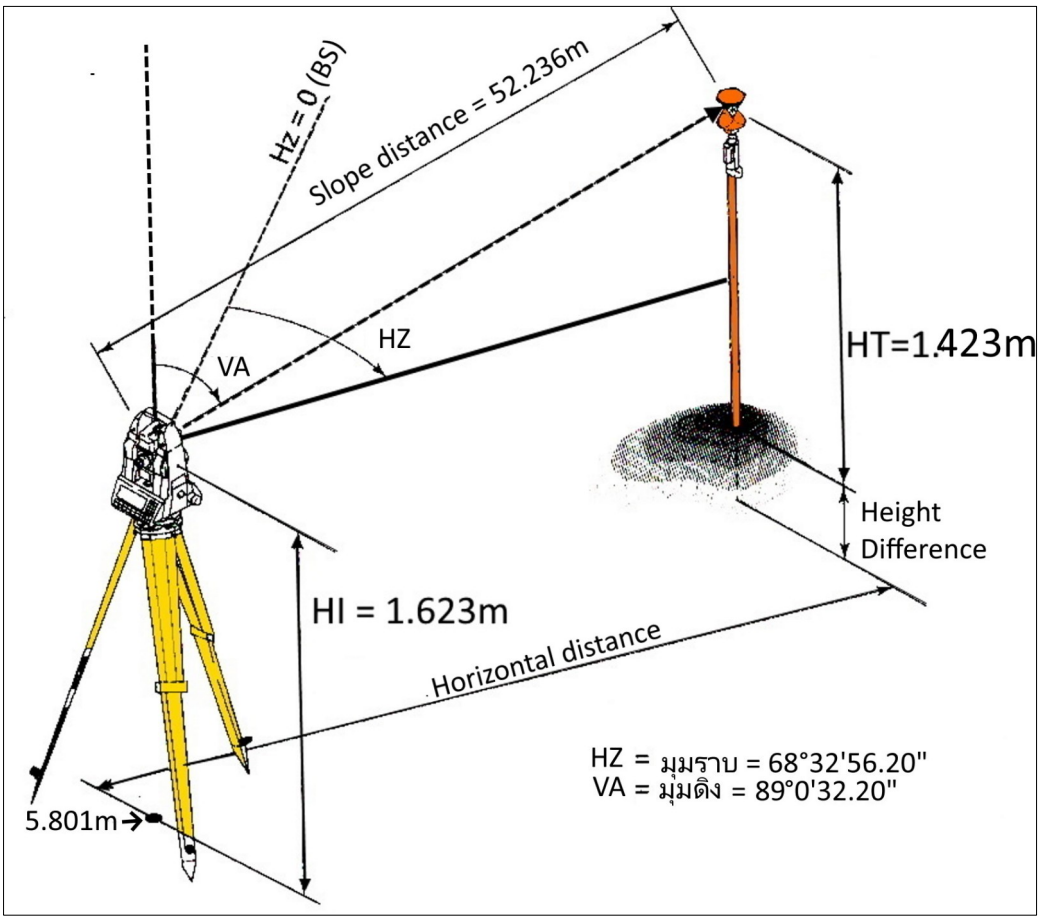


ป้อนค่า Coordinate 3D

โปรแกรมจะคำนวณอะซิมัท ระยะทางจากจุดตั้งกล้องไปเป้าหลัง กดคีย์ “EXE” จะได้ผลลัพธ์ อะซิมัท ระยะราบทั้งระยะบนพื้นโลกและระยะกริดจากจุดตั้งกล้องไปเป้าหน้า สุดท้ายคือผลลัพธ์ที่ต้องการคือค่าพิกัด **N=2632215.761, E=232911.782** และค่าระดับ = 6.905 เมตร ของเป้าหน้า



ผลลัพธ์ Coordinate 3D



ไดอะแกรมแสดงการวัดหาค่าพิกัดและค่าระดับจากตัวอย่าง

สรุป

ผมเขียนโปรแกรมพื้นฐานงานสำรวจไว้หลายชุด จะทยอยมาลงเป็นซีรี่ย์ ผมเห็นว่ารุ่น **fx-9750GIII** ราคาประมาณสามพันบาทนั้นไม่ถือว่าแพงมาก สำหรับช่างสำรวจหรือช่างโยธาที่มีการงานเป็นหลักแหล่งสามารถหาซื้อมาใช้งานได้ น้ำหนักเบาจับถนัดมือ

ช่องทางการโปรแกรมมิ่งมีสามช่องทาง ช่องทางที่ 1 คือเขียนด้วยภาษาเบสิคคาสิโอ ก็คล้ายๆกับเครื่องคิดเลขรุ่น fx-5800 อาจจะต่างกันนิดหน่อย ช่องทางที่ 2 คือเขียนด้วยภาษาซี ต้องลงเครื่องมือช่วยพัฒนาโปรแกรม SDK ของคาสิโอ ช่องทางนี้ยากสุดครับ ช่องทางที่ 3 คือเขียนด้วยภาษาไพทอน ที่ผมนำเสนอในบทความนี้ ซึ่งก็ไม่ยากใครมีพื้นฐานโปรแกรมมิ่งภาษาไพทอนนิดๆหน่อยๆก็เขียนได้ ตอนหน้ามาต่อกับ**โปรแกรมพื้นฐานงานสำรวจชุดที่ 2 (COGO Selected Serie 2)** ว่าด้วยเรื่องวงกลมล้วนๆ

มีข้อเตือนใจสำหรับช่างสำรวจสักนิด ปัจจุบันโปรแกรมบนกล้องประมวลผลรวม (Total station) ก็เก่งขึ้นเรื่อยๆ ไม่ต้องกดเครื่องคิดเลขเหมือนสมัยแต่ก่อน ยังมีแอพบนโทรศัพท์มือถือมาช่วยด้วย แต่พื้นฐานจริงๆคือความเข้าใจในเรื่องเรขาคณิต (Geometry) ค่าพิกัด ระยะทาง มุมอะซิมัท จะเกี่ยวข้องกันอย่างแยกกันไม่ออก ดังนั้นต้องทำความเข้าใจ

หลักวิชาสำรวจให้ลึกซึ้ง เมื่อเข้าใจดีแล้ว การใช้โปรแกรมบนเครื่องคิดเลขหรือบนแอพโทรศัพท์มือถือก็เป็นแค่เครื่องมือที่มาช่วยเหลือให้คำนวณผลลัพธ์มาได้เร็วและถูกต้อง

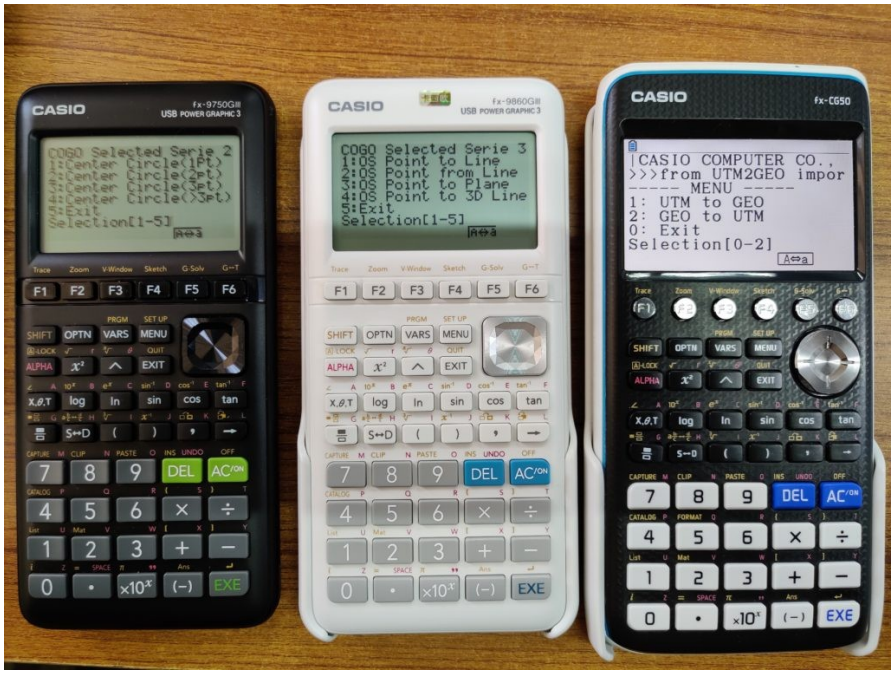
อนาคตอันใกล้หุ่นยนต์ ปัญญาประดิษฐ์จะมาเร็วกว่าที่คิด หุ่นยนต์หรืออะไรที่มันทำงานแทนเราได้อัตโนมัติ ตัวอย่างคนทำแผนที่ทางอากาศจากโดรนไม่ต้องรู้เรื่อง photogrammetry เลยก็ทำได้ แต่ถ้าคนเราเองไม่ช่วยตัวเองศึกษาค้นคว้า บอกได้คำเดียวว่าเสร็จหุ่นยนต์แน่นอน

วิธีการติดตั้งโปรแกรมบนเครื่องคิดเลข

ผมจะปล่อยซอร์สโค้ดที่อัพเดทไว้ที่เมนูดาวน์โหลด (Download) เมื่อดาวน์โหลดไฟล์ “**COGOSSE1.py**” และไฟล์ “**pbrutils.py**” แล้ว สมมุติโหลดมาไว้ที่เครื่องคอมพิวเตอร์ จากนั้นนำเครื่องคิดเลขคาสิโอ fx-9750GIII หรือ fx-9860GIII หรือ fx-cg50 มาเสียบเข้าสาย USB ต่อกับเครื่องคอมพิวเตอร์ ให้กดคีย์ “F1” เพื่อจำลองเป็นไดรว์ จากนั้น copy สองไฟล์นี้เข้าไปที่ไดรว์ของเครื่องคิดเลข รายละเอียดอ่านที่ลิ๊งค์ที่ผมจัดทำมาเป็นพิเศษ

โปรแกรมภาษาไพทอนบนเครื่องคิดเลขคาสิโอ fx-9750GIII fx-9860GIII และ fx-cg50 โปรแกรมพื้นฐานงานสำรวจชุดที่ 2 (COGO Selected Serie 2)

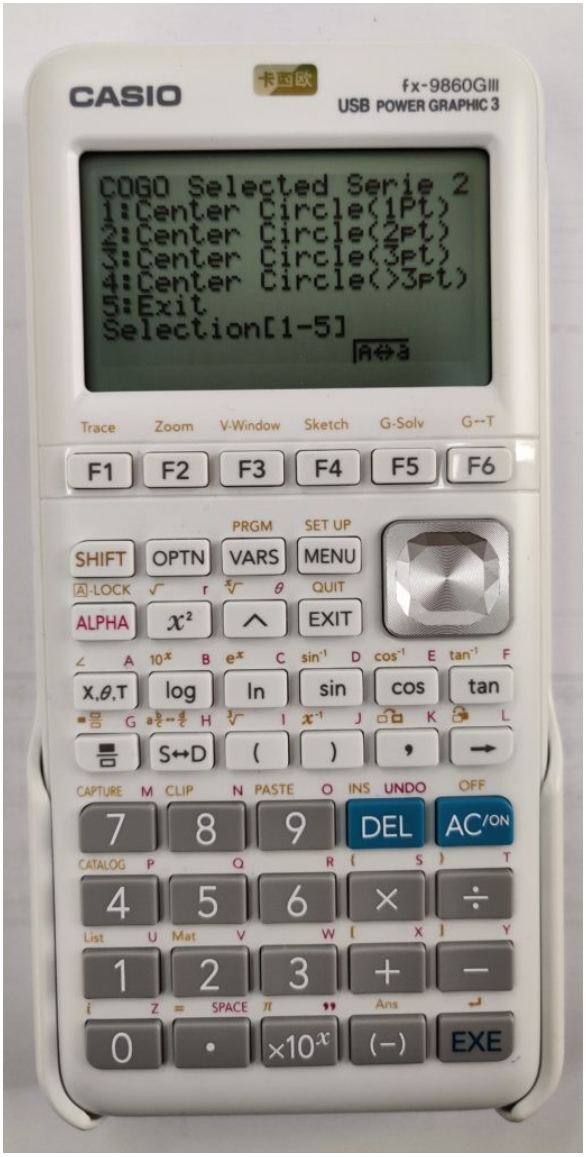
ตอนนี้มีเครื่องคิดเลขของคาสิโอสามรุ่นที่สามารถโปรแกรมด้วยภาษาไพทอนหรือไมโครไพทอน (MicroPython) ได้ คือ fx-9750GIII, fx-9860GIII และ fx-cg50 ข้อดีของภาษาไพทอนนั้นคือง่าย ทรงพลัง แต่ข้อจำกัดของเครื่องคิดเลขคือหน่วยความจำที่มีมาน้อย ดังนั้นบนเครื่องคิดเลขจะมีไลบรารีที่นำมาจากเครื่องคอมพิวเตอร์มาใช้งานได้น้อย ต้องปรับกันพอสมควร ไม่มีไลบรารีเทพแบบ Numpy ที่จะมาใช้คำนวณเรื่องเมตริกซ์ (Matrix) ดังนั้นถ้าใช้เมตริกซ์ก็ต้องออกแรงเขียนโค้ดเองมากหน่อย แต่ยังมี Matplotlib ฉบับย่อที่พอกล้อมแกล้มได้เล็กน้อย



เครื่องคิดเลขคาสิโอ fx-9750GIII, fx-9860GIII และ fx-cg50

ชุดโปรแกรมพื้นฐานงานสำรวจ COGO (Coordinate Geometry) ชุดที่ 2

มาถึงโปรแกรมพื้นฐานงานสำรวจชุดที่ 2 (COGO Selected Serie 2) ซีรี่ย์ว่าด้วยเรื่องวงกลมล้วนๆ การประยุกต์ใช้บางงานเจอเสากลมที่ก่อสร้างไว้แล้ว เราต้องไปเก็บ as-built เพื่อหาขนาดเสาและค่าพิกัดจุดศูนย์กลาง หรือถ้าเป็นงานก่อสร้างเช่นเจาะเสาเข็มนำร่องด้วยเคสสิ้ง (casing) ลักษณะเป็นวงกลมตรงกลางกลวงเพื่อจะสั่นด้วยไวเบรเตอร์ทะลุเข้าไปในดิน ช่างสำรวจก็จะเก็บจุดด้านบนของเคสสิ้ง 3 จุด เพื่อหาจุดศูนย์กลางแล้วเทียบกับค่าพิกัดที่ออกแบบไว้ว่ามีการเบี่ยงเบนไปเท่าไหร่ ถ้ามากเกินข้อกำหนด จะต้องมีการถอนเคสสิ้งเพื่อปักใหม่



เครื่องคิดเลขคาสิโอ fx-9860GIII รันโปรแกรมพื้นฐานงานสำรวจชุดที่ 2

ส่วนประกอบของโปรแกรม

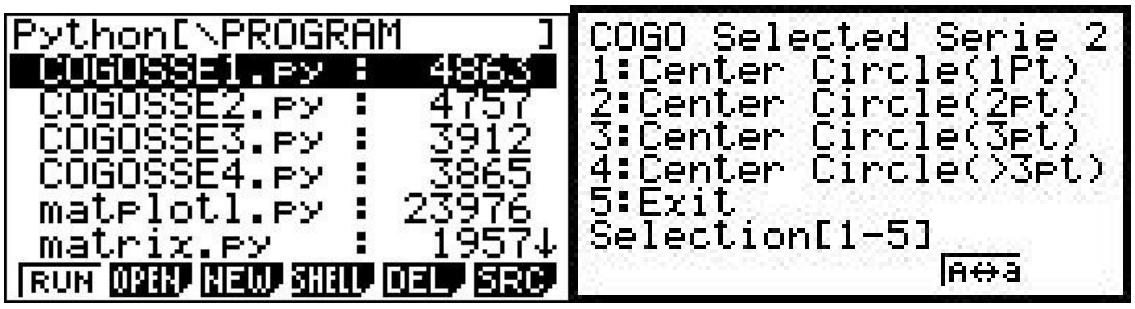
สำหรับโปรแกรมพื้นฐานงานสำรวจในชุดนี้จะจัดโปรแกรมย่อยเล็กๆ ไว้ 4 โปรแกรม

1. **Center Circle(1Pt)** เมื่อทราบค่าพิกัด 1 จุดบนเส้นรอบวง ในภาคสนามจะหาค่าพิกัดจุดศูนย์กลางของวงกลม as-built ของเสา โดยการตั้งกล้องแล้วเล็งเป้าหลังมาแตะที่ขอบวงกลมด้านซ้ายวัดมุมและกวาดมาแตะขอบวงกลมด้านขววัดมุม ป้อนตัวเลขโปรแกรมจะบอกมุมให้เล็งมาที่กลางเสาทำการวัดระยะทาง โปรแกรมจะคำนวณค่าพิกัดจุดศูนย์กลางของวงกลมและรัศมีของเสาให้
2. **Center Circle(2Pt)** เมื่อทราบค่าพิกัด 2 จุด บนเส้นรอบวง และทราบรัศมี สามารถคำนวณหาค่าพิกัดจุดศูนย์กลางได้ แต่จะมีวงกลมสองวงที่ผ่านจุดนี้ ดังนั้นคำตอบจะมีสองค่า

3. **Center Circle(3Pt)** เมื่อทราบค่าพิกัด 3 จุด บนเส้นรอบวง ในภาคสนามที่ประยุกต์ใช้กันจะเป็นลักษณะงานก่อสร้างเจาะเสาเข็มที่ผมกล่าวไว้ข้างต้น ช่างสำรวจวัดค่าพิกัดบนเส้นรอบวงของเคสสิ้ง กะระยะให้ห่างเท่าๆกันทั้งสามจุด ทำมุมสามเหลี่ยมประมาณ 60 องศาเพื่อลดความคลาดเคลื่อน
4. **Center Circle(>3pt)** เมื่อทราบค่าพิกัดมากกว่า 3 จุด ในกรณีนี้การคำนวณจะใช้วิธี **least squares** มาช่วยในการคำนวณ ตัวอย่างการประยุกต์ใช้งานเช่นการเก็บแท๊งค์วงกลมขนาดใหญ่ในโรงกลั่นแก๊สหรือน้ำมัน เราจะเก็บจุดมารอบๆ อาจจะ 10 จุดหรือถึง 100 จุดเพื่อมาคำนวณหารูปวงกลมที่ฟิต (Circle fit) ดีที่สุด ในสมัยนี้การวัดระยะทางแบบ Reflector-less ทำได้ง่าย

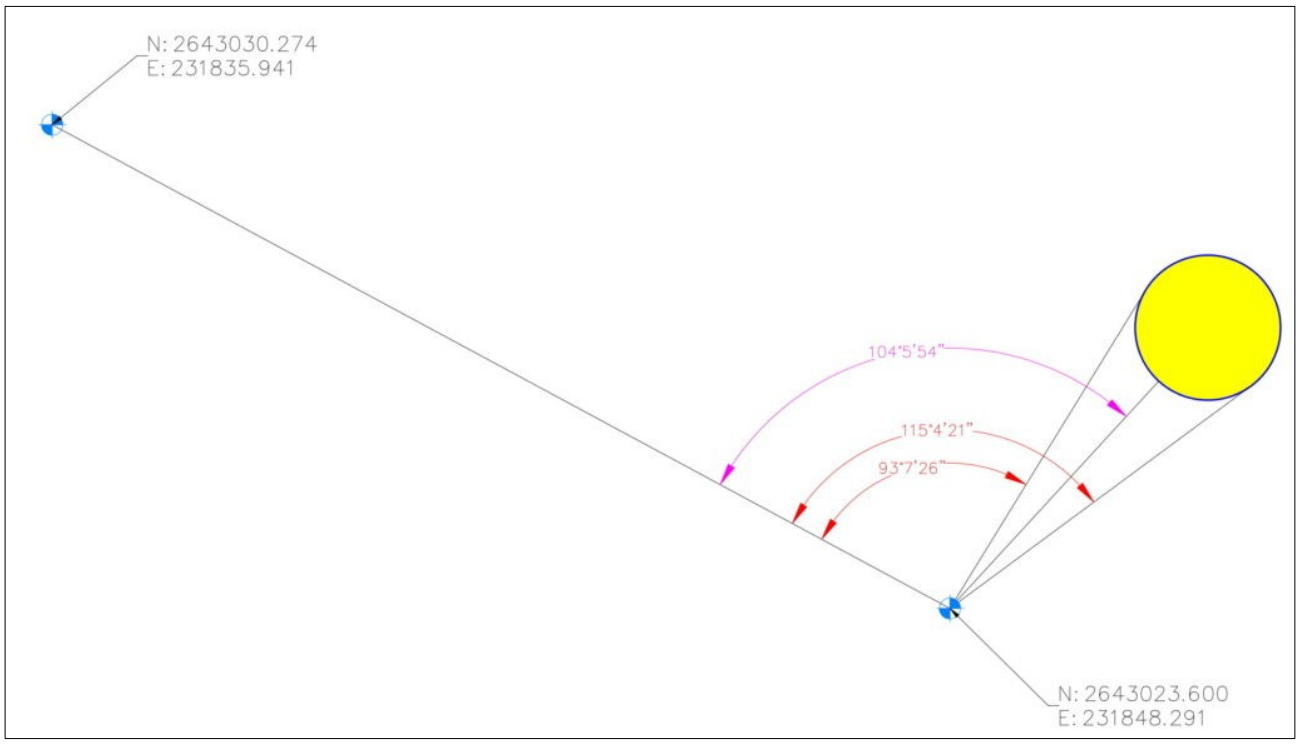
วิธีการใช้งานโปรแกรม

กดคีย์ “**MENU**” ที่เครื่องคิดเลข กดคีย์ลูกศรเลื่อนไปที่ไอคอน “Python” กดเข้าไป ตรงนี้ผมสร้างโฟลเดอร์ชื่อ “PROGRAM” ผมเอาซอร์สโค้ดโปรแกรมไพทอนมาไว้ที่นี่ เมื่อกด “EXE” เข้าไปจะเห็นไฟล์ซอร์สโค้ดไพทอนเรียงกันประมาณนี้ ตอนนี้จะรันไฟล์ COGOSSE2.py ซึ่งจะต้องมีไฟล์ไลบรารี pbrutils.py ด้วย



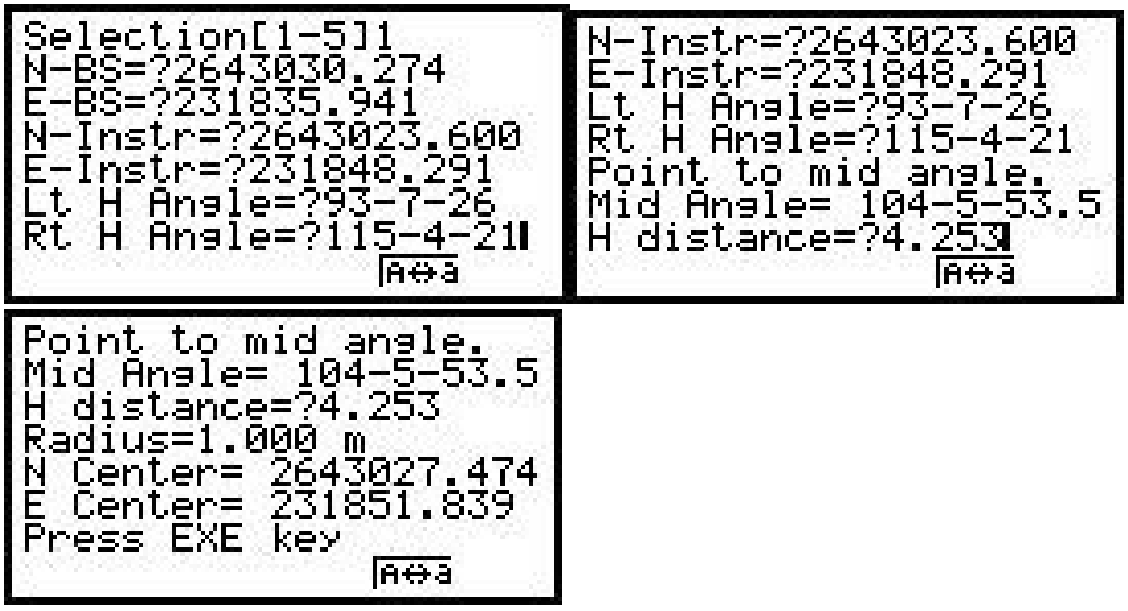
Center Circle(1Pt)

มาดูตัวอย่างจากรูปเพื่อให้เข้าใจได้ง่าย จุดตั้งกล้องมีค่า N=2643023.600, E=231848.291 เป้าหลัง N=2643030.274, E=231835.941 วัดมุมไปที่จุดสัมผัสเสาเข็ม(สีเหลือง) 93° 7′ 26″ แล้ววัดมุมไปที่จุดสัมผัสเสาเข็มด้านขวาอ่านมุมได้ 115° 4′ 21″ เปิดมุมไปตรงกลางเสา (มุมคำนวณโดยโปรแกรม) 104° 5′ 54″ วัดระยะทางได้ 4.253 เมตร หาค่าพิกัดจุดศูนย์กลางเสาเข็มและคำนวณหารัศมี



ตัวอย่าง Center Circle (1 point)

กดคีย์ “1” และกดคีย์ “EXE” ป้อนค่าตามรูปด้านล่าง การป้อนมุมให้คีย์เครื่องหมายลบ (-) คั่น ไม่สามารถใช้รูปแบบป้อนมุมแบบเครื่องคิดเลขได้เพราะคาสิโอไม่เปิดฟังก์ชั่นให้ใช้งาน ในภาคสนามหลังจากป้อนมุมสัมผัสด้านซ้ายและด้านขวาโปรแกรมจะคำนวณมุมให้ 104° 5′ 53.5″ แล้วทำการวัดระยะทาง

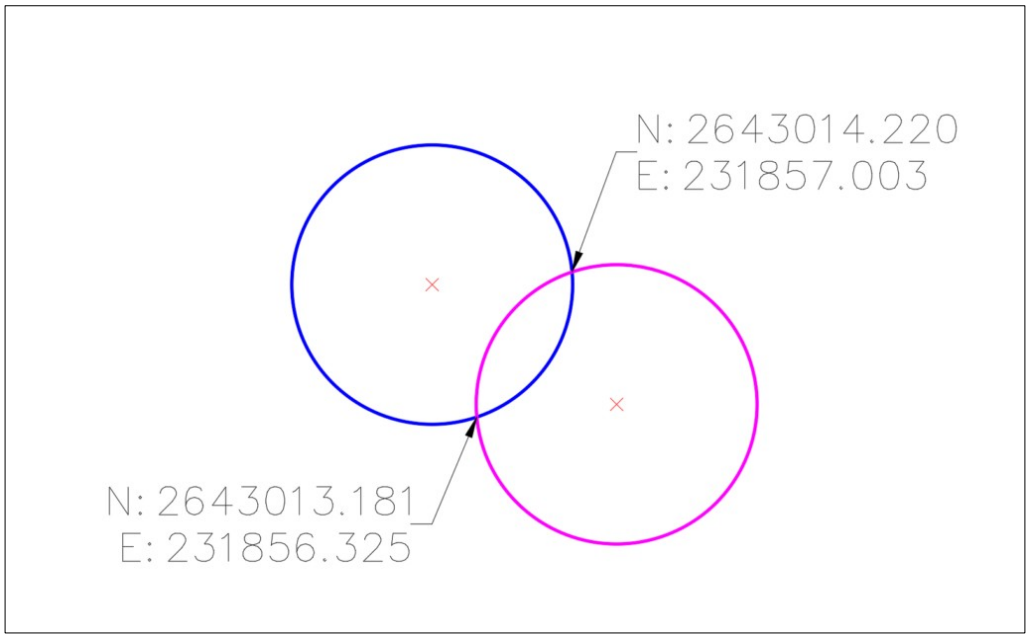


Center Circle (1 point)

จะได้ค่าพิกัดจุดศูนย์กลางของเสาเข็ม N=2643027.474, E=231851.839 รัศมีเสาเข็ม 1.0 เมตร

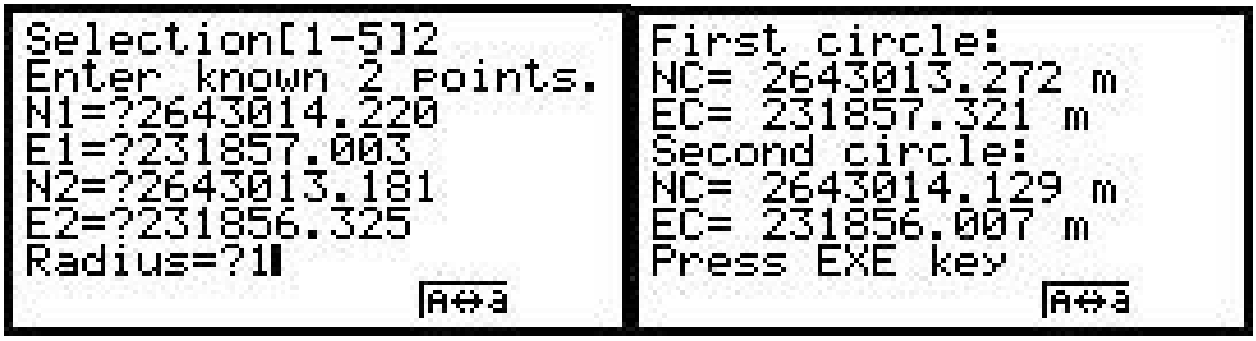
Center Circle(2Pt)

ในกรณีที่ทราบค่าพิกัดของจุดบนเส้นรอบวงกลม 2 จุด และทราบค่ารัศมีมาลอง จุดแรก N=2643014.220, E=231857.003 จุดที่ 2 N=2643013.181, E=231856.325 รัศมีวงกลม 1 เมตร



ตัวอย่าง Center Circle (2 points)

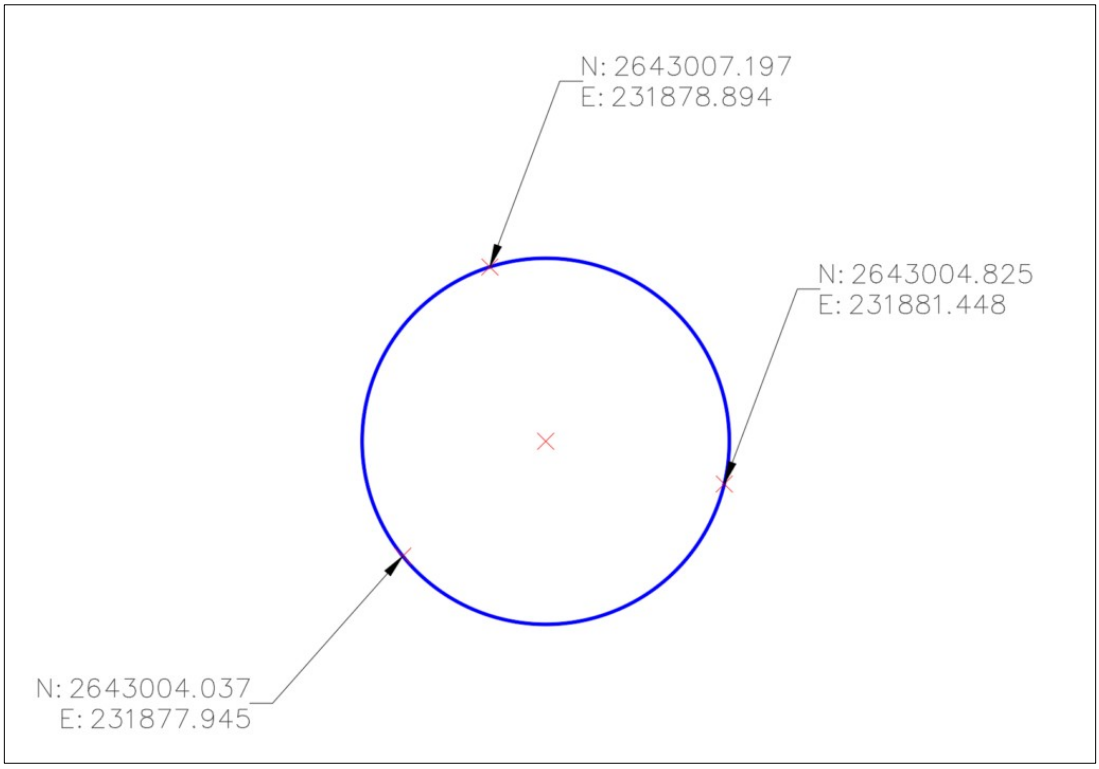
จากเมนูกดคีย์ “2” และกดคีย์ “EXE” ป้อนค่า จะได้ผลลัพธ์ออกมาเป็นวงกลมสองวงคือวงกลมสีม่วงมีค่าพิกัดที่ศูนย์กลางวงกลมคือ N=2643013.272, E=231857.321 วงกลมที่ 2 คือวงกลมสีน้ำเงิน N=2643014.129, E=231856.007



Center Circle (2 points)

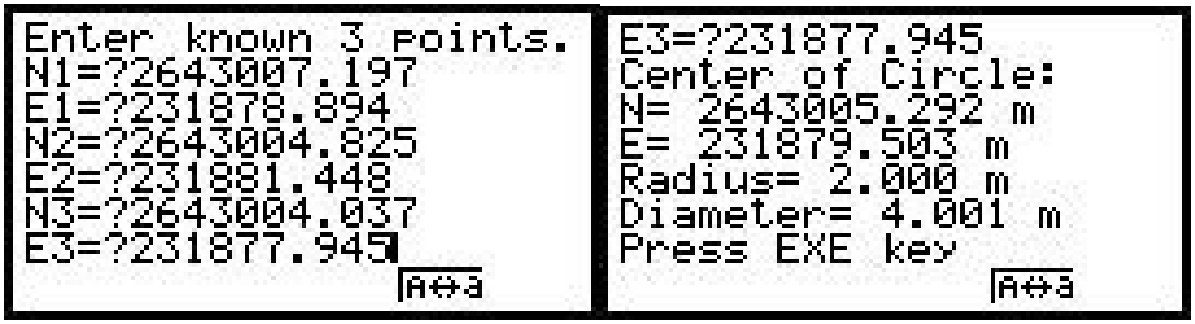
Center Circle(3Pt)

ในกรณีทราบค่าพิกัดจุดบนเส้นรอบวง 3 จุด จะสามารถหาค่าพิกัดจุดศูนย์กลางและรัศมีของวงกลมได้ ตัวอย่าง ค่าพิกัดจุดที่ 1 N=2643007.197, E=231878.894 ค่าพิกัดจุดที่ 2 N=2643004.825, E=231881.448 ค่าพิกัดจุดที่ 3 N=2643004.037, E=231877.94



ตัวอย่าง Circle Center (3 points)

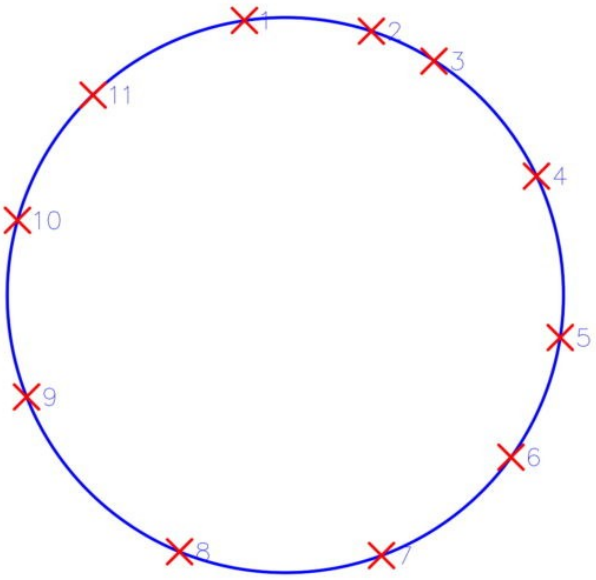
จากเมนูกดคีย์ “3” และกดคีย์ “EXE” ป้อนค่า จะได้ผลลัพธ์ออกมาค่าพิกัดจุดศูนย์กลาง ตลอดจนรัศมีวงกลมและขนาดเส้นผ่านศูนย์กลาง ค่าพิกัดของจุดศูนย์กลางคือ N=2643005.292, E=231879.503 รัศมี = 2.000 เมตร เส้นผ่าศูนย์กลาง 4.001 เมตร



Center circle (3 points) การประยุกต์ใช้งานในการหาตำแหน่งปักเคสสิ้งเสาเข็ม

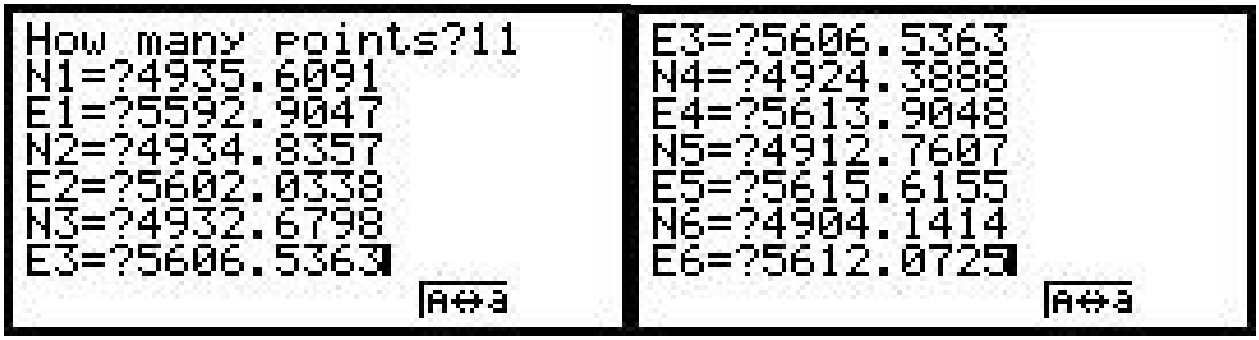
Center Circle(>3pt)

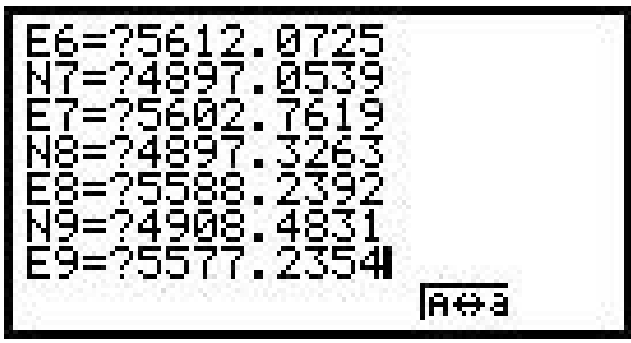
ถ้าจุดบนเส้นรอบวงเกิน 3 จุด การคำนวณจะยุ่งยากขึ้นมาทันที ในกรณนี้เราจะต้องใช้วิธีคำนวณแบบ least squares ในบางครั้งการคำนวณแบบนี้จะเรียกว่า Circle Fit by least squares วิธีอื่นที่ผมเห็นมาคือวิธี Circle Fit by algebraic ผมขอเน้นแบบ least squares เพราะว่าคุ้นเคยและให้ค่าดีกว่า ผมไปดัดแปลงโค้ดภาษาซีเป็นภาษาไพทอนจากลิ๊งค์นี้ เครดิต: Mark J. Kilgard มาดูตัวอย่าง

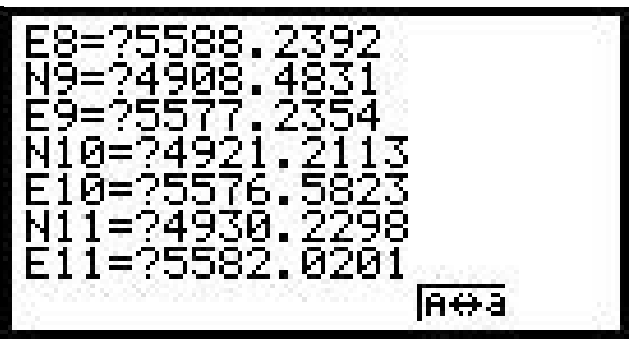


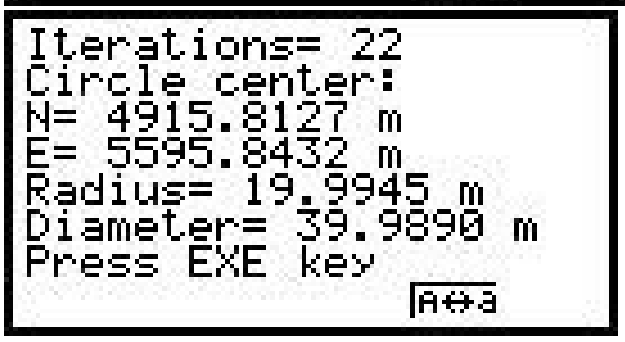
No.	N	E
1	4935.6091	5592.9047
2	4934.8357	5602.0338
3	4932.6798	5606.5363
4	4924.3888	5613.9048
5	4912.7607	5615.6155
6	4904.1414	5612.0725
7	4897.0539	5602.7619
8	4897.3263	5588.2392
9	4908.4831	5577.2354
10	4921.2113	5576.5823
11	4930.2298	5582.0201

จากเมนูกดคีย์ “4” และกดคีย์ “EXE” โปรแกรมจะถามจำนวนจุดตอบ จากตัวอย่างมีทั้งหมด 11 จุด จากนั้น โปรแกรมจะเริ่มถามค่าพิกัดจุดที่1 ก็ป้อน N1=4935.6091, E1=5592.9047 จนถึงจุดที่ 11 N11=4930.2298, E11=5582.0201 จากนั้นโปรแกรมก็คำนวณค่าพิกัดให้จากวิธี least squares วนรอบ 22 ครั้งถึงได้คำตอบ จะได้ค่าพิกัดศูนย์กลางที่ฟิตที่สุด N=4915.8127, E=5595.8432 รัศมี 19.9945 เมตร ถ้าขนาดจริงในแบบก่อสร้างรัศมี 20 เมตร ต่างกันประมาณ 5 มม. อาจเป็นไปได้ว่าก่อสร้างไม่ได้ตามแบบเป๊ะหรือไม่อาจจะมีการบิดรูปของโครงสร้าง (deformation) ก็ต้องศึกษากันต่อไป



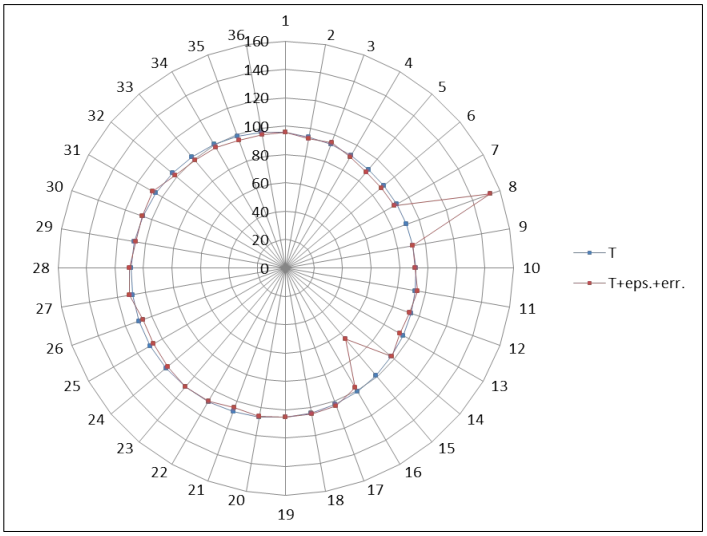






สรุป

สำหรับโปรแกรมย่อยที่ 4 ผมไม่ได้ตรวจหา error สมมติว่าจุดที่กระโดดออกไม่เข้ากลุ่มอาจจะเกิดจากความผิดพลาดตอนสำรวจหรือคีย์ข้อมูลผิดดูตัวอย่างภาพด้านล่างจะเห็นจุดที่ 8 และจุดที่ 15 ผมจะเพิ่มเติมให้ในอนาคต ส่วนโปรแกรมย่อยจาก 1 ถึง 3 นั้นเป็นพื้นฐานมากๆ โปรดติดตามโปรแกรมพื้นฐานงานสำรวจชุดที่ 3 ต่อไป (COGO Serie 3) ว่าด้วยเรื่องการ offset จุด



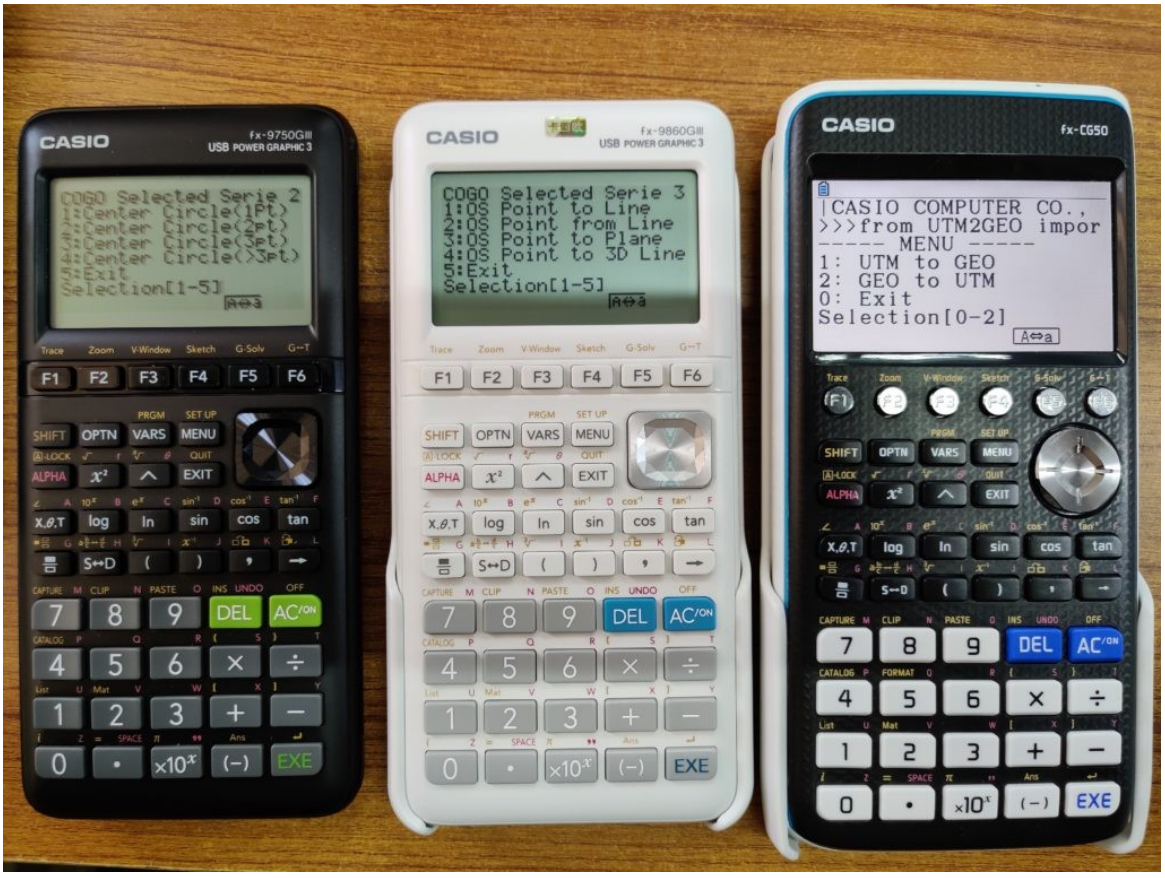
เครดิตภาพ: stackoverflow.com

วิธีการติดตั้งโปรแกรมบนเครื่องคิดเลข

ผมจะปล่อยซอร์สโค้ดที่อัพเดทไว้ที่เมนูดาวน์โหลด (Download) เมื่อดาวน์โหลดไฟล์ “**COGOSSE2.py**” และไฟล์ “**pbrutils.py**” แล้ว สมมุติโหลดมาไว้ที่เครื่องคอมพิวเตอร์ จากนั้นนำเครื่องคิดเลขคาสิโอ fx-9750GIII หรือ fx-9860GIII หรือ fx-cg50 มาเสียบเข้าสาย USB ต่อกับเครื่องคอมพิวเตอร์ ให้กดคีย์ “F1” เพื่อจำลองเป็นไดรว์ จากนั้น copy สองไฟล์นี้เข้าไปที่ไดรว์ของเครื่องคิดเลข รายละเอียดอ่านที่ลิ๊งค์ที่ผมจัดทำมาเป็นพิเศษ

โปรแกรมภาษาไพทอนบนเครื่องคิดเลขคาสิโอ fx-9750GIII fx-9860GIII และ fx-cg50 โปรแกรมพื้นฐานงานสำรวจชุดที่ 3 (COGO Selected Serie 3)

ตอนนี้มีเครื่องคิดเลขของคาสิโอสามรุ่นที่สามารถโปรแกรมด้วยภาษาไพทอนหรือไมโครไพทอน (MicroPython) ได้ คือ fx-9750GIII, fx-9860GIII และ fx-cg50 ทั้งสามรุ่นสามารถหาซื้อได้ไม่ยากนัก ผมซื้อมาทางออนไลน์สะดวกดี สนนราคาเรียงตามรุ่นตอนนี้อยู่ที่ สามพันบาท สี่พันห้าร้อยบาทและเจ็ดพันกว่าบาทตามลำดับ ผมแนะนำให้สำหรับคนที่เบี้ยน้อยหอยน้อยลงทุนกับ fx-9750GIII เพราะราคาไม่แพง คุ้มค่าเกินราคา ทั้งสามรุ่นสามารถโปรแกรมด้วยภาษาคาสิโอเบสิคและภาษาไพทอน แล้วแต่ความถนัด ความชอบ

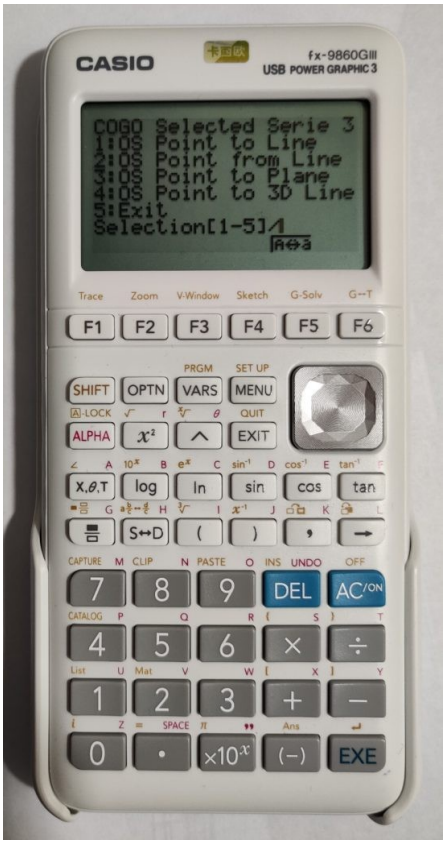


เครื่องคิดเลขคาสิโอ fx-9750GIII, fx-9860GIII และ fx-cg50

ถ้าโปรแกรมด้วยไพทอนจะสามารถเขียนโปรแกรมที่ยากๆหรือคณิตศาสตร์ซับซ้อนขึ้นมาได้ นี่เป็นเหตุผลสำคัญ ถึงแม้หน่วยความจำของเครื่องคิดเลขจะน้อยก็เป็นโจทย์ให้ผู้พัฒนาโปรแกรมได้ออปติไมซ์โปรแกรมให้ทำงานได้ แต่สำหรับโปรแกรมพื้นฐานงานสำรวจส่วนใหญ่ก็พื้นๆ ไม่มีอะไรซับซ้อน

ชุดโปรแกรมพื้นฐานงานสำรวจ COGO (Coordinate Geometry) ชุดที่ 3

มาถึงโปรแกรมพื้นฐานงานสำรวจชุดที่ 3 (COGO Selected Serie 3) ซี่รี่ย์ว่าด้วยเรื่องการ offset ของจุดและเส้นตรง การประยุกต์งานเช่นบางสถานการณ์ไปวางจุดหน้างานแต่ทำไม่ได้เนื่องจากตำแหน่งที่วางจุดมีอุปสรรคจำเป็นต้องเลื่อนหรือออฟเซ็ทไป สามารถใช้เครื่องคิดเลขคิดได้ทันที



โปรแกรมพื้นฐานงานสำรวจชุดที่ 3 บนเครื่องคิดเลขคาสิโอ fx-9860GIII

ส่วนประกอบของโปรแกรม

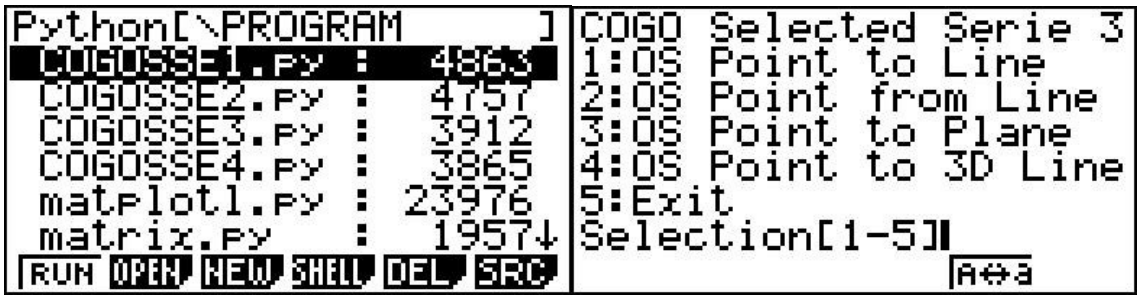
สำหรับโปรแกรมพื้นฐานงานสำรวจในชุดนี้จะจัดโปรแกรมย่อยเล็กๆ ไว้ 4 โปรแกรม เนื่องจากหน้ากว้างของหน้าจอเครื่องคิดเลขมีแค่ 21 ตัวอักษร ดังนั้นบางอย่างผมใช้คำย่อเช่น OS ย่อมาจาก Offset

1. **OS Point to Line** เมื่อรู้ค่าพิกัดของจุดและรู้เส้นตรงจากค่าพิกัด 2 จุด สามารถหาระยะ offset จากจุดไปตั้งฉากกับเส้นตรงได้
2. **OS Point from Line** เมื่อกำหนดเส้นตรงจากค่าพิกัด 2 จุด (จุดที่ 1 และจุดที่ 2) และต้องการ offset จุดจากเส้นตรงโดยที่ทราบระยะทางจากจุดที่ 1 (Chainage) และระยะ offset สามารถ offset จุดไปทางซ้ายหรือทางขวาได้ จากนั้นสามารถคำนวณหาค่าพิกัดของจุด

3. **OS Point to Plane** มีระนาบที่กำหนดจากจุด 3 จุดที่ทราบค่า (X, Y, Z) และต้องการหาระยะทางที่สั้นที่สุดระยะหว่างระนาบและจุดที่ทราบค่า (X, Y, Z)
4. **OS Point to 3D Line** เมื่อกำหนดเส้นตรง 2 จุด ที่ทราบค่า (X, Y, Z) ต้องการหาระยะ offset ที่สั้นที่สุดระหว่างจุดที่ทราบค่า (X, Y, Z)

วิธีการใช้งานโปรแกรม

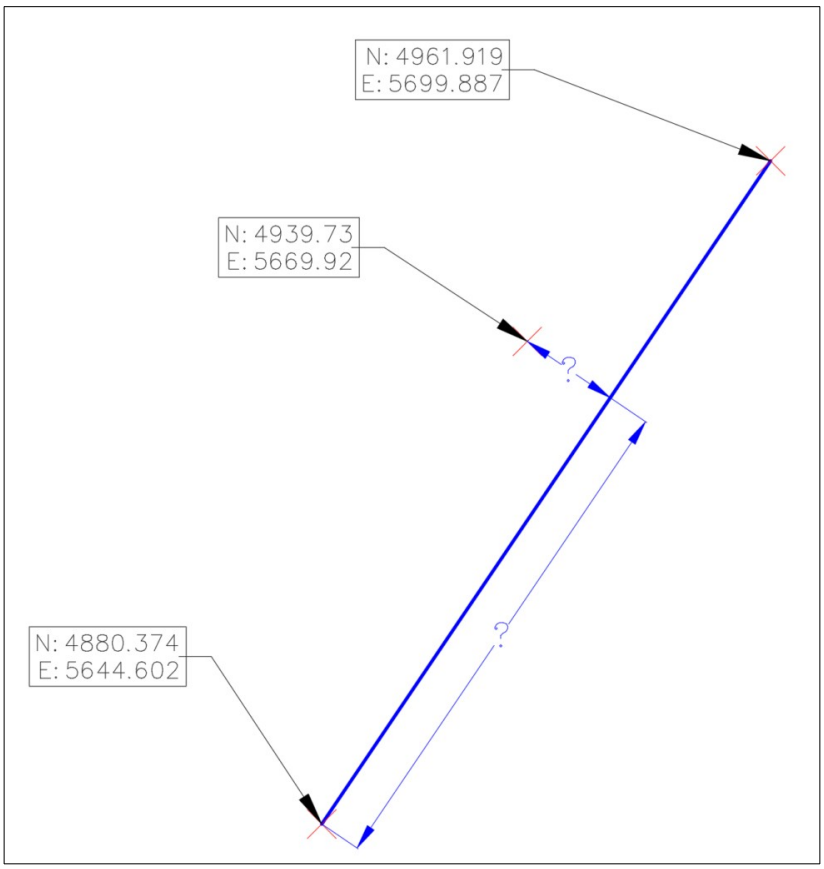
กดคีย์ “**MENU**” ที่เครื่องคิดเลข กดคีย์ลูกศรเลื่อนไปที่ไอคอน “Python” กดเข้าไป ตรงนี้ผมสร้างโฟลเดอร์ชื่อ “PROGRAM” ผมเอาซอร์สโค้ดโปรแกรมไพทอนมาไว้ที่นี่ เมื่อกด “EXE” เข้าไปจะเห็นไฟล์ซอร์สโค้ดไพทอนเรียงกันประมาณนี้ ตอนนี้จะรันไฟล์ COGOSSE3.py ซึ่งจะต้องมีไฟล์ไลบรารี pbrutils.py ด้วย



ซอร์สโค้ดไพทอนและเมนูโปรแกรมพื้นฐานงานสำรวจ ชุดที่ 3 (COGO Selected Serie 3)

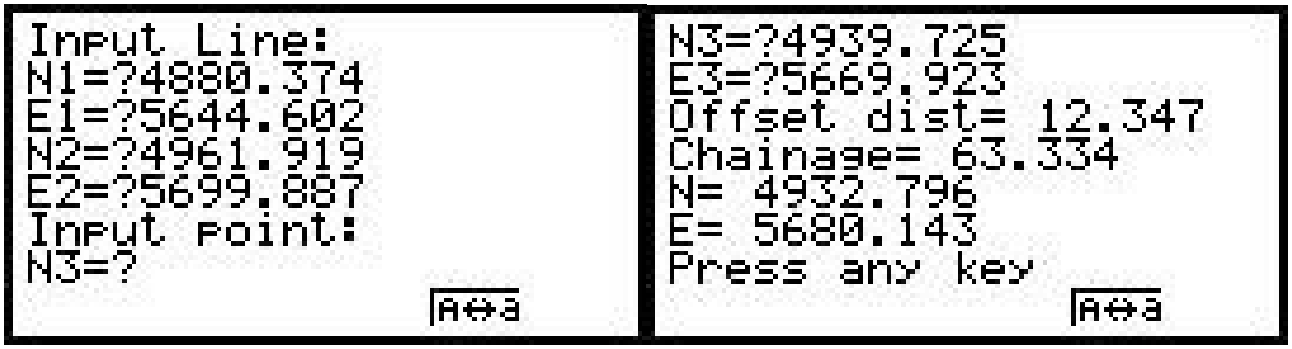
OS Point to Line

ตัวอย่างกำหนดค่าพิกัดเส้นตรงจุดที่ 1 N=4880.374, E=5644.602 จุดที่ 2 N=4961.919, E=5699.887 กำหนดจุด N=4939.725, E=5669.923 หาระยะทาง (offset) จากจุดนี้ไปตั้งฉากกับเส้นตรงและคำนวณหาระยะทางจากเส้นตรงจุดที่ 1 (chainage)



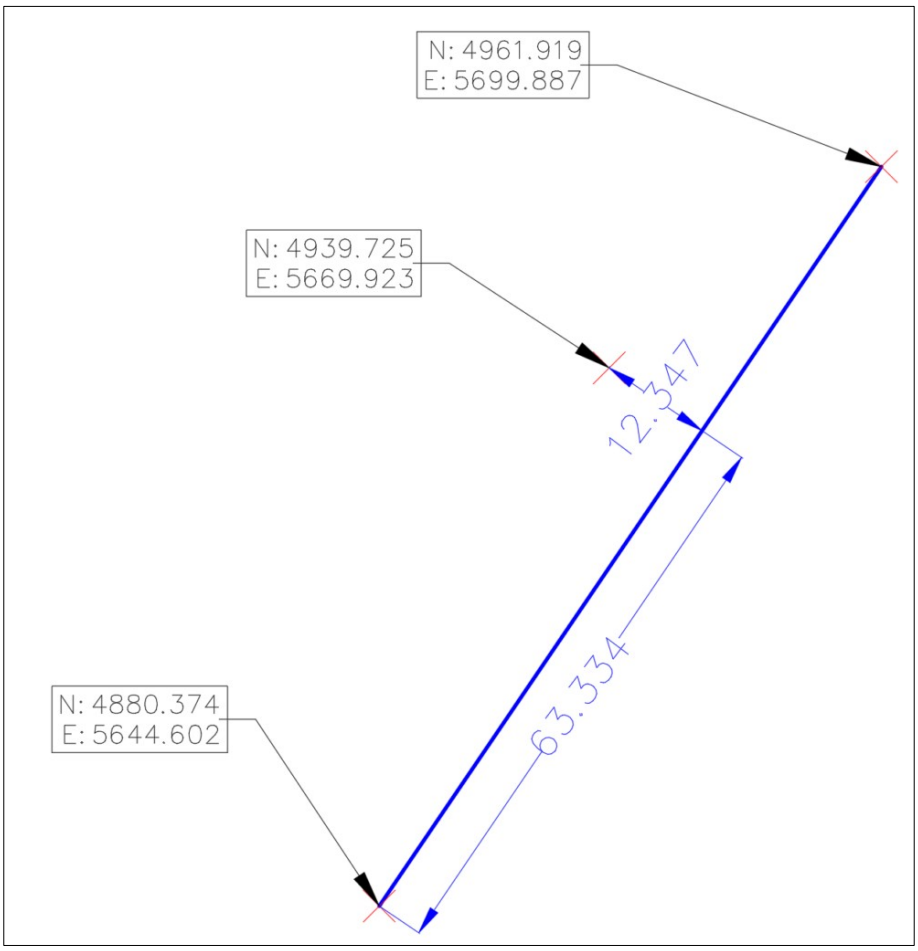
แผนผังแสดงตัวอย่าง OS Point to Line (2D)

ที่เมนูกดคีย์ “1” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัดจุดที่ 1 และจุดที่ 2 จากนั้นจะถามค่าพิกัดจุดที่ต้องการ offset ไปหาเส้นตรง



การป้อนค่าและผลลัพธ์การคำนวณ OS Point to Line (2D)

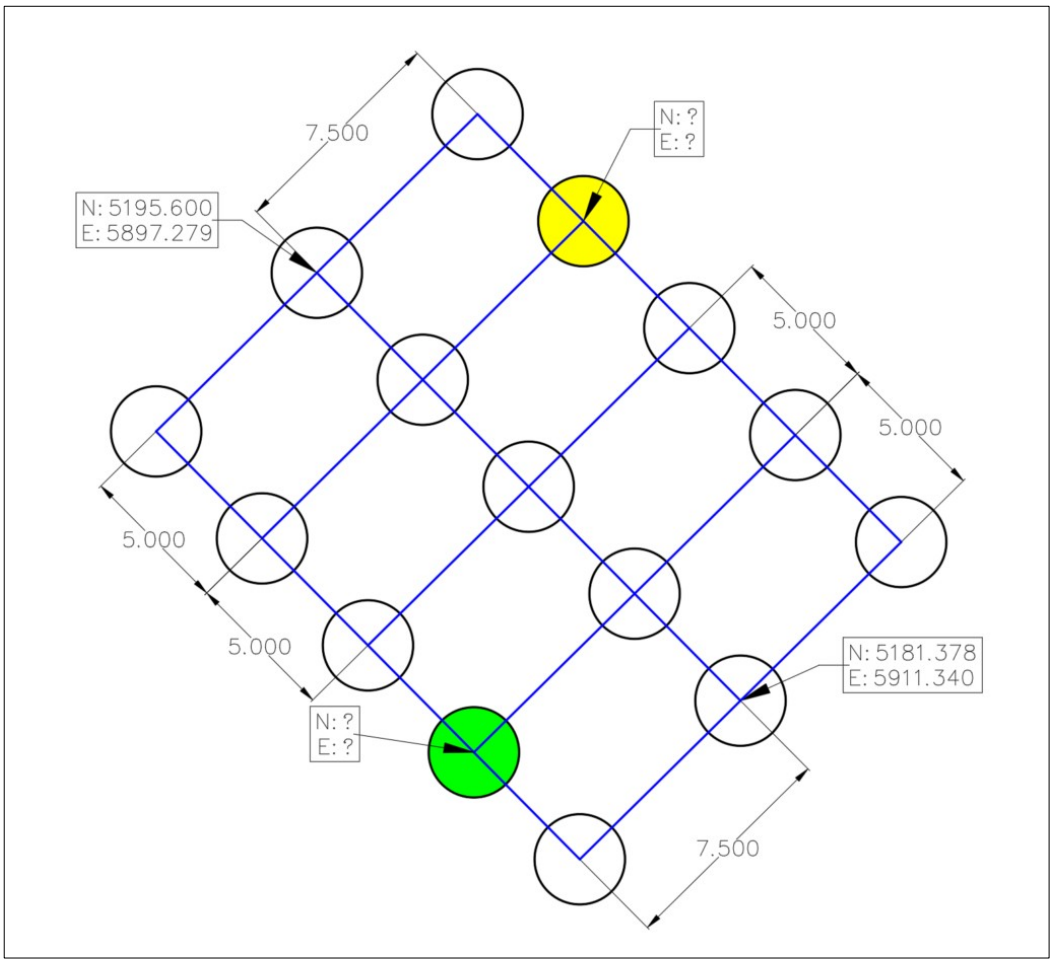
จะได้ระยะทาง (offset) จากจุดไปตั้งฉากกับเส้นตรงเท่ากับ 12.347 และระยะทาง (chainage) จากเส้นตรงจุดที่ 1 เท่ากับ 63.334



แผนผังแสดงผลลัพธ์ OS Point to Line (2D)

OS Point from Line

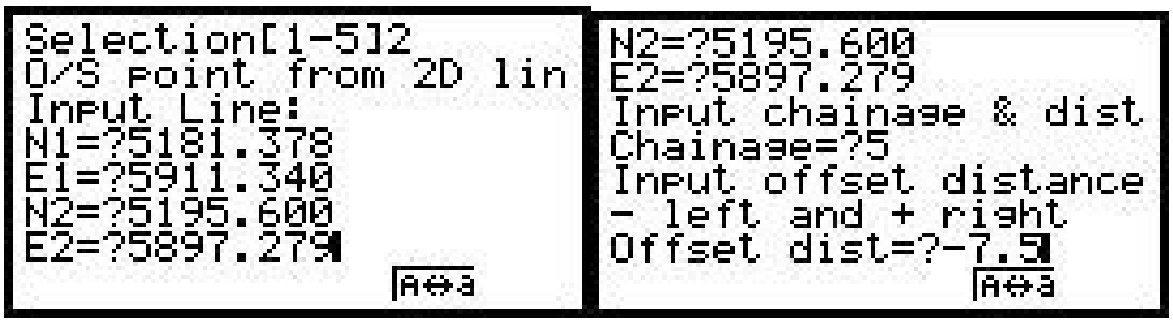
เมื่อกำหนดเส้นตรงที่รู้ค่าพิกัด 2 จุด จากนั้นกำหนดระยะทาง (chainage) จากเส้นตรงจุดที่ 1 และกำหนดระยะ offset ไปทางซ้าย (ป้อนเครื่องหมายเป็นลบ) หรือไปทางขวา ต้องการหาค่าพิกัดของจุดที่ offset ตัวนี้ มาดูตัวอย่าง

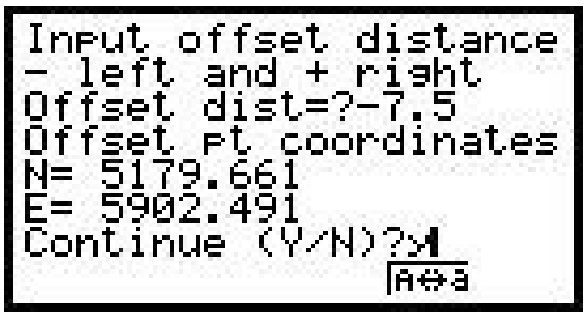


แผนผังแสดงตัวอย่างกลุ่มเสาเข็มที่รู้ค่าพิกัด 2 จุด

กำหนดค่าพิกัดของกลุ่มเส้นตรงโดยที่จุดที่ 1 N=5181.378, E=5911.340 และจุดที่ 2 N=5195.600, E=5897.279 ต้องการทราบค่าพิกัดของเสาเข็มสีเขียวและสีเหลือง

ที่เมนูกดคีย์ “2” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัดจุดที่ 1 และจุดที่ 2 หาค่าพิกัดเสาเข็มสีเขียวได้จากระยะทาง (chainage) จากจุดที่ 1 ไป 5 เมตร และระยะ offset ไปทางซ้ายเท่ากับ -7.5 เมตร

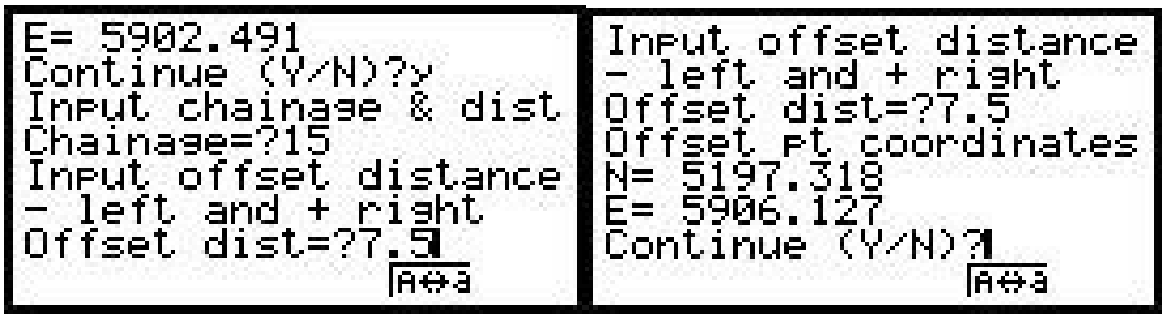




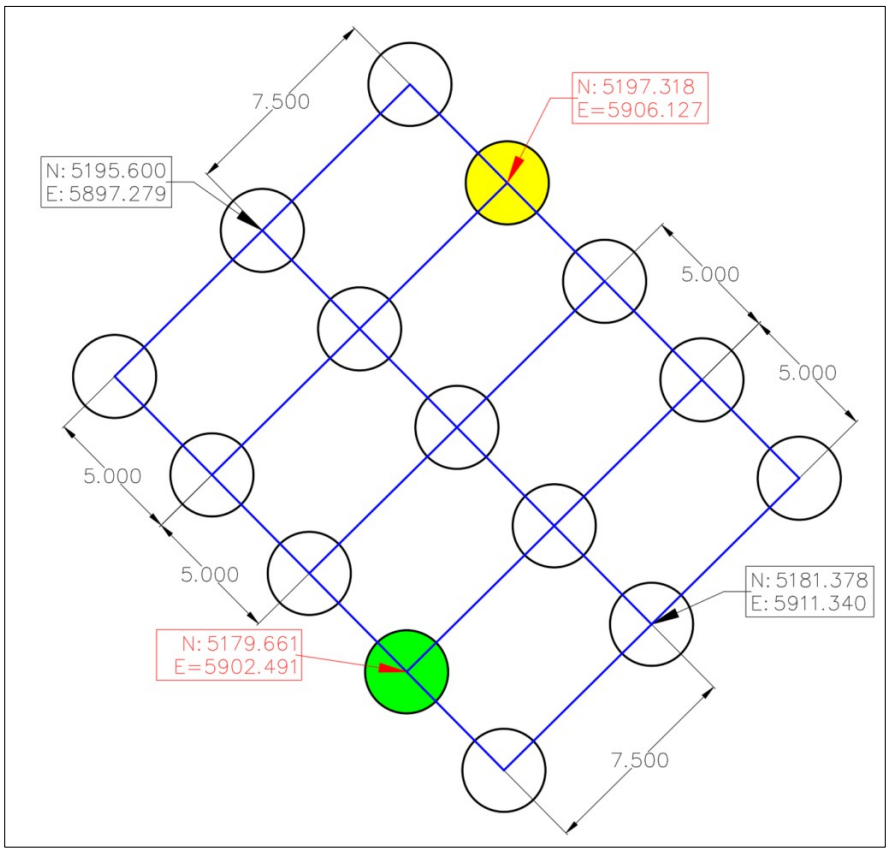
การป้อนค่าและคำนวณหาค่าพิกัดเสาเข็มสีเขียว (2D)

จะได้ค่าพิกัดเสาเข็มสีเขียว N=5179.661, E=5902.491 จะคำนวณค่าพิกัดเสาเข็มสีเหลือง โปรแกรมจะถามว่าต้องการคำนวณต่อหรือไม่ “Continue (Y/N)?” ที่เครื่องคิดเลขให้กดคีย์ “ALPHA” แล้วกดคีย์ “-” เพื่อให้ออกเป็นตัว “y”

หาพิกัดเสาเข็มสีเหลืองโดยการป้อน chainage = 3×5=15 และค่า offset = 7.5 จะได้ค่าพิกัดเสาเข็มสีเหลือง N=5197.318, E=5906.127 เมื่อโปรแกรมว่าคำนวณต่ออีกหรือไม่ “Continue (Y/N)?” ตอนนี้กดคีย์ “EXE” เพื่อไม่คำนวณต่อ



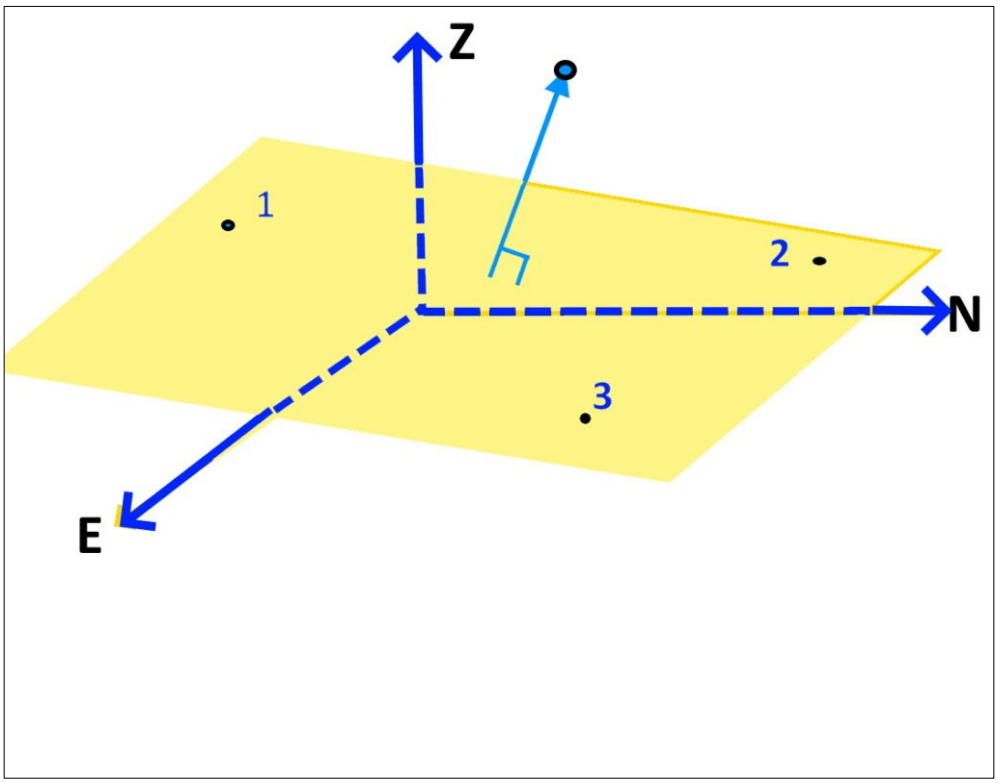
การคำนวณหาค่าพิกัด offset เสาเข็มสีเหลือง (2D)



แผนผังแสดงผลลัพธ์ค่าพิกัดผลการคำนวณ

OS Point to Plane

บางกรณีต้องทำงานบนพื้นที่ที่เป็นระนาบ (plane) เอียง ตัวอย่างเช่นการติดตั้งโบลท์บนพื้นที่เป็นระนาบเอียงต้องติดตั้งเพื่อให้ตั้งฉากกับระนาบ เวลาหาความสูงของโบลท์หรือระยะทางตั้งฉากจากจุดไปยังระนาบเอียง กรณีที่ทราบค่าพิกัด (X, Y, Z) ของระนาบซึ่งจะต้องทราบค่า 3 จุด เมื่อรังวัดหัวโบลต์ด้วยกล้องประมวลผลรวม (total station) จะทราบค่าพิกัดและค่าระดับหัวโบลท์ เมื่อนำมาคำนวณสามารถทราบความสูงของโบลท์จากระนาบได้



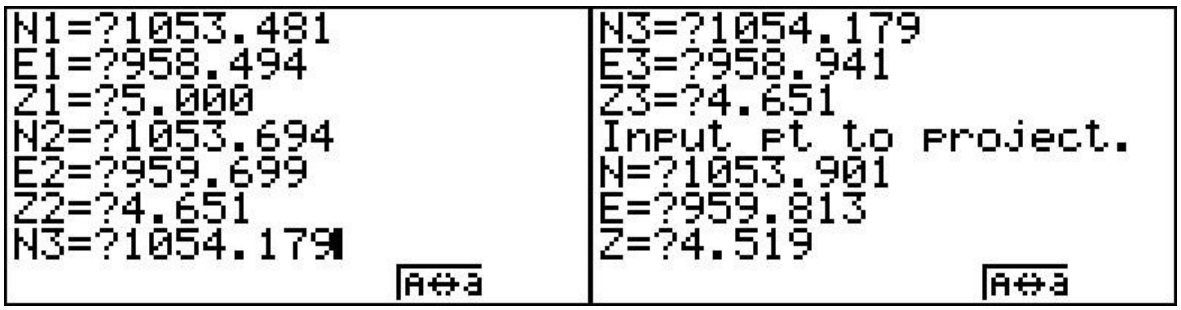
ไดอะแกรมแสดงจุดกับระนาบ (plane) ในระบบพิกัดสามมิติ (3D)

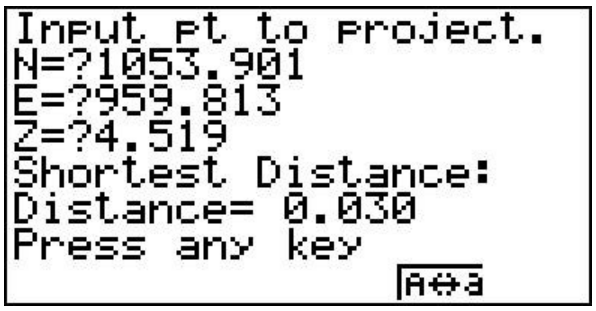
ที่เมนูกดคีย์ “3” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัด 3D ของระนาบ จุดที่ 1, จุดที่ 2 และจุดที่ 3 จากนั้นป้อนจุด 3D โปรแกรมจะคำนวณหาระยะทางจากจุดมาตั้งฉากกับระนาบ (ระยะทางที่สั้นที่สุด)

กำหนดจุดบนระนาบ (plane) จำนวน 3 จุดดังนี้

1. จุดที่ 1 N=1053.481, E=958.494, Elev=5.000 m
2. จุดที่ 2 N=1053.694, E=959.699, Elev=4.651 m
3. จุดที่ 3 N=1054.179, E=958.941, Elev=4.651 m

กำหนดจุดเซอร์เวย์ที่ได้จากรังวัด N=1053.901, E=959.813, Elev=4.519 m คำนวณหาระยะตั้งฉากจากจุดนี้ไปยังระนาบ





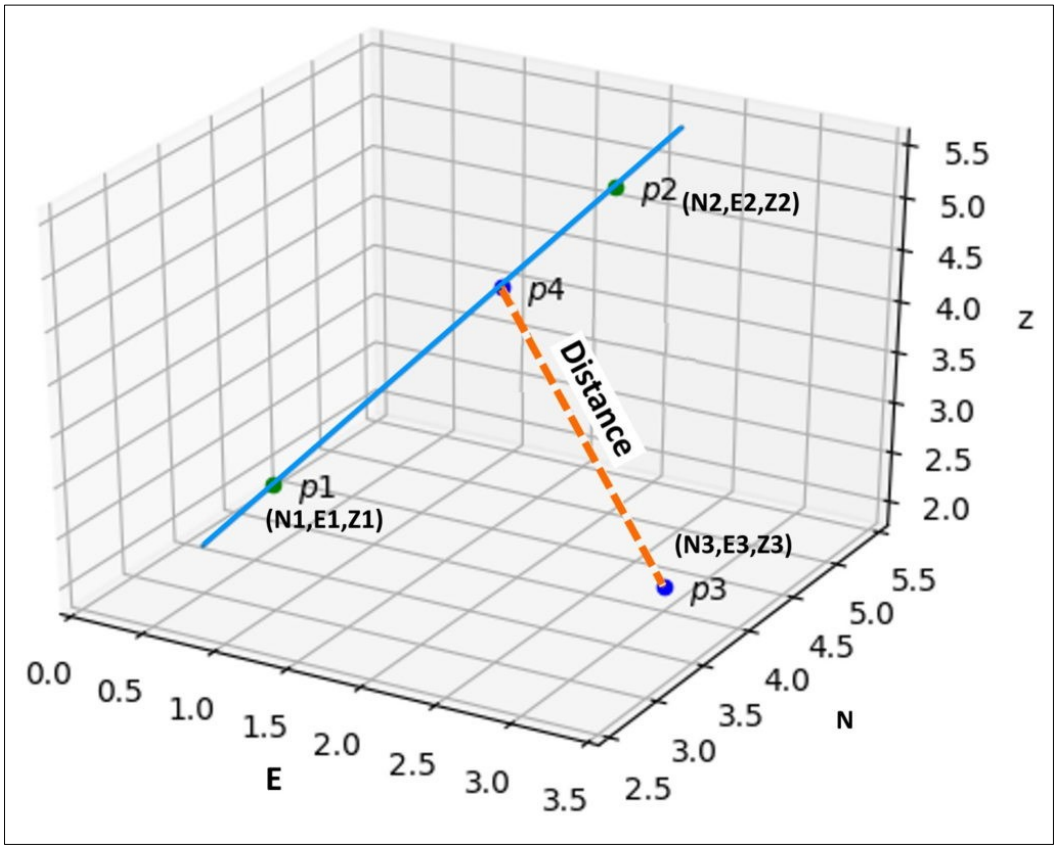
การป้อนค่าและผลลัพธ์ OS Point to Plane (3D)

คำตอบระยะทางจากจุดมาตั้งฉากกับระนาบ = 0.030 m = 30mm

OS Point to 3D Line

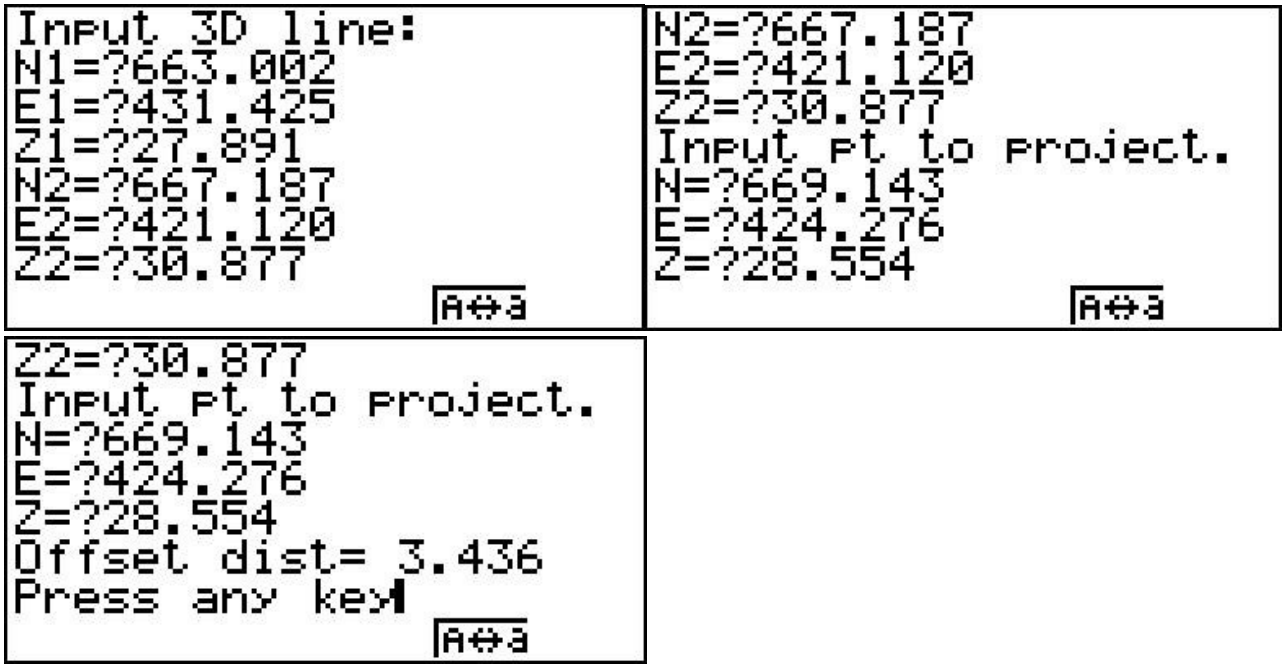
ตัวอย่างต้องการคำนวณหาระยะทาง offset จากจุด p3 N=669.143, E=424.276, Elev=28.554 ไปหาเส้นตรงที่กำหนดค่าพิกัดและระดับดังนี้

1. จุด p1N=663.002, E=431.425, Elev=27.891
2. จุด p2 N=667.187, E=421.120, Elev=30.877



ไดอะแกรมแสดงเส้นตรง p1-p2 ในระบบสามมิติกับระยะทางที่สั้นที่สุดไปหาจุด p3

ที่เมนูกดคีย์ “4” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัด 3D ของเส้นตรง จุดที่ 1, จุดที่ 2 จากนั้นจะถามค่าพิกัดและระดับของจุด โปรแกรมจะคำนวณหาระยะทางจากจุดมาตั้งฉากกับเส้นตรง 3D (ระยะทางที่สั้นที่สุด)



การป้อนค่าและผลลัพธ์ OS Plane to Line (3D)

คำตอบระยะ offset = 3.436

สรุป

การทำงานสำรวจนั้นพื้นฐานก็คือเรื่องเรขาคณิตและตรีโกณมิติ เกี่ยวพันเรื่องระยะทางและมุม เมื่อบวกเรื่องค่าระดับก็จะกลายเป็นงานในระบบสามมิติ (3D) ไม่จำเป็นต้องไปจำสูตรให้ยุ่งยาก ให้รู้ที่มาที่ไปของสูตรก็พอ เครื่องคิดเลขก็เป็นเครื่องมืออย่างหนึ่ง ถ้าจำสูตรไม่ได้ก็ต้องใช้โปรแกรมมาช่วย ผมตั้งใจจะเขียนโปรแกรมพื้นฐานงานสำรวจอีกหลายชุดทยอยตามกันมา ก็หวังว่าจะเป็นประโยชน์ไม่มากก็น้อยและก็เหมือนเดิมคือแจกฟรี ไม่คิดสตางค์ ตั้งใจเพื่อช่วยเหลือวงการสำรวจบ้านเราเล็กๆน้อยๆที่พอทำได้ สำหรับซอร์สโค้ดโปรแกรมพื้นฐานงานสำรวจชุดที่ 3 ก็เอามาแปะไว้ด้านล่าง

วิธีการติดตั้งโปรแกรมบนเครื่องคิดเลข

ผมจะปล่อยซอร์สโค้ดที่อัพเดทไว้ที่เมนูดาวน์โหลด (Download) เมื่อดาวน์โหลดไฟล์ “**COGOSSE3.py**” และไฟล์ “**pbrutils.py**” แล้ว สมมุติโหลดมาไว้ที่เครื่องคอมพิวเตอร์ จากนั้นนำเครื่องคิดเลขคาสิโอ fx-9750GIII หรือ fx-9860GIII หรือ fx-cg50 มาเสียบเข้าสาย USB ต่อกับเครื่องคอมพิวเตอร์ ให้กดคีย์ “F1” เพื่อจำลองเป็นไดรว์ จากนั้น copy สองไฟล์นี้เข้าไปที่ไดรว์ของเครื่องคิดเลข รายละเอียดอ่านที่ลิ๊งค์ที่ผมจัดทำมาเป็นพิเศษ

โปรแกรมภาษาไพทอนบนเครื่องคิดเลขคาสิโอ fx-9750GIII fx-9860GIII และ fx-cg50 โปรแกรมพื้นฐานงานสำรวจชุดที่ 4 (COGO Selected Serie 4)

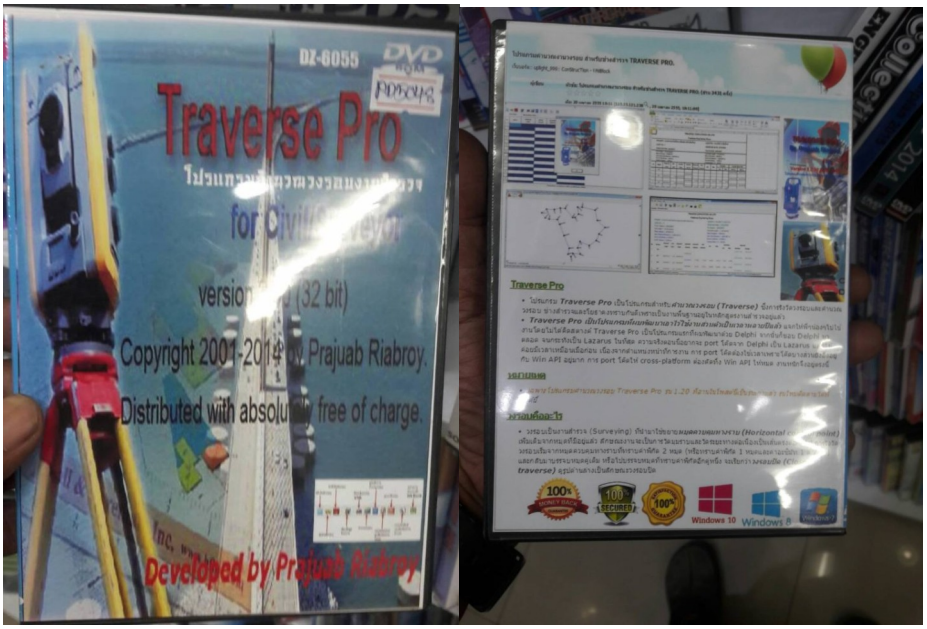
ตอนนี้มาถึงโปรแกรมพื้นฐานงานสำรวจชุดที่ 4 (COGO Selected Serie 4) สำหรับเครื่องคิดเลขคาสิโอ fx-9750GIII, fx-9860GIII และ fx-cg50 PRIZM สามรุ่นที่รองรับภาษาไพทอนหรือไมโครไพทอน ได้ในขณะนี้ หาซื้อได้ในเมืองไทย ราคาย่อมเยาที่สุดคือ fx-9750GIII ที่ราคาประมาณสามพันบาท ถ้ามีงานการทำเป็นหลักเป็นแหล่งแล้วไม่น่าแพง บางทีเราซื้อโทรศัพท์มือถือได้ราคาเป็นเรือนหมื่นไม่คิดอะไรมาก แต่กับเครื่องคิดเลขคิดแล้วคิดอีก สำหรับผมแล้ว ราคาขนาดนี้ถือว่าคุ้มค่ามาก ผมไปนั่งพลิกดูด้านหลังเครื่อง fx-9860GIII ปรากฏเห็น “Made in Thailand” ค่อนข้างประหลาดใจพอสมควร พอไปพลิกดูเครื่อง fx-cg50 กลับเป็น “Made in China”

เครื่องคิดเลขคาสิโอ fx-9750GIII, fx-9860GIII และ fx-CG50 PRIZM

วิธีการติดตั้งโปรแกรมไพทอนลงบนเครื่องคิดเลข

ผมเคยเห็นแผ่นผีลงโปรแกรมคำนวณวงรอบ “**Traverse Pro**” ที่ศูนย์ไอทีหลายแห่งเมื่อประมาณ 7-8 ปีที่แล้ว ก็สงสัยอยู่ครามครันว่าโปรแกรมมันแจกฟรีในบล็อกผมอยู่แล้ว ทำไมคนยังต้องซื้อแผ่นผีอีก ก็มานึกได้ว่าในยุคนั้นสังคมโซเชียลยังไม่เบ่งบานเท่าปัจจุบัน คนที่ไม่รู้ก็ไม่มีเครือข่ายบอกกันต่อ เช่นเดียวกัน ผมคิดว่าการก็อปปี้โปรแกรมภาษาไพทอนลงเครื่องคิดเลขเป็นเรื่องง่ายๆ ที่ใครๆเข้าใจได้ คงจะไม่ใช่ เลยจัดทำวิธีการติดตั้งลงโปรแกรมภาษาไพทอนลง

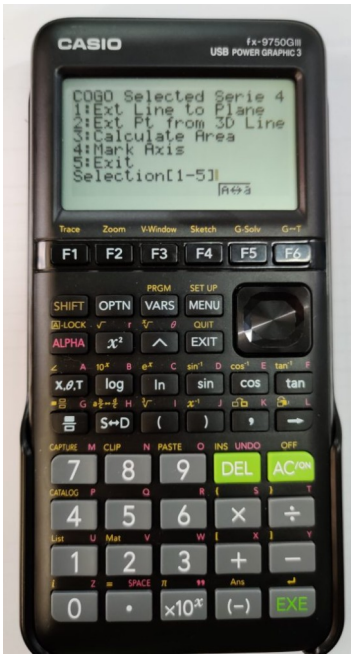
เครื่องคิดเลข ขั้นตอนก็ไปดาวน์โหลดโปรแกรมจากบล็อกนี้เข้าเครื่องคอมพิวเตอร์ จากนั้นเอาสายเคเบิ้ลมินิยูเอสบี มาเสียบเครื่องคอมพิวเตอร์และเครื่องคิดเลขจากนั้นทำตามคู่มือลิ๊งค์นี้ครับ



ย้อนความทรงจำโปรแกรม Traverse Pro ยุคสมัยที่ได้รับเกียรติถูกปั๊มเป็นแผ่นผีขายตามศูนย์ไอที

ชุดโปรแกรมพื้นฐานงานสำรวจ COGO (Coordinate Geometry) ชุดที่ 4

มาถึงโปรแกรมพื้นฐานงานสำรวจชุดที่ 4 (COGO Selected Serie 4) ซี่รี่ย์ว่าด้วยเรื่องการยืดจุดเส้นตรงในสามมิติไปยังระนาบ (plane) หรือการยืดจุดจากเส้นตรงสามมิติเมื่อทราบระยะทาง ตลอดจนการคำนวณหาพื้นที่ (area) และสุดท้ายคือโปรแกรมมาร์คแกน (mark axis) ซึ่งรายละเอียดมาดูกันอีกทีว่าใช้งานและการประยุกต์ใช้งานอย่างไรบ้าง



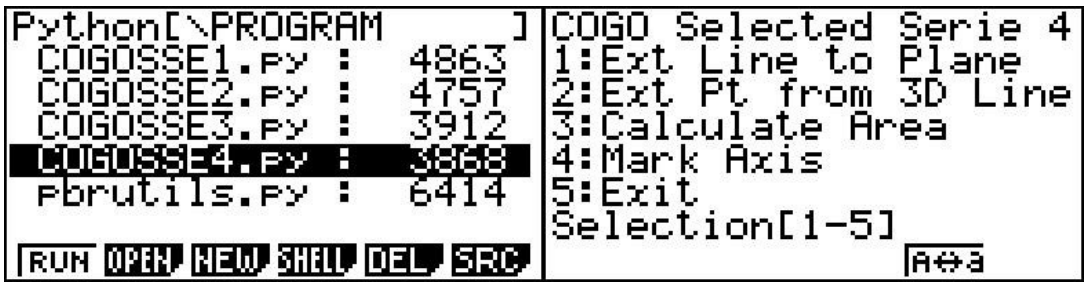
ส่วนประกอบของโปรแกรม

สำหรับโปรแกรมพื้นฐานงานสำรวจในชุดนี้จะจัดโปรแกรมย่อยเล็กๆ ไว้ 4 โปรแกรม เนื่องจากหน้ากว้างของหน้าจอเครื่องคิดเลขมีแค่ 21 ตัวอักษร ดังนั้นบางอย่างผมใช้คำย่อเช่น Ext ย่อมาจาก Extend แปลว่ายืดหรือขยาย และคำว่า Pt ย่อมาจาก point คือจุดนั่นเอง

1. **Ext Line to Plane** เมื่อรู้ค่าพิกัดของเส้นตรง (X,Y,Z) 2 จุด และระนาบที่กำหนดด้วยจุด (X,Y,Z) 3 จุด สามารถยืดเส้นตรงไปแตะระนาบแล้วหาจุดพิกัด (X,Y,Z) นั้นได้
2. **Ext Pt from 3D Line** เมื่อรู้ค่าพิกัดของเส้นตรง (X,Y,Z) 2 จุด ต้องการยืดระยะทางไปตามที่กำหนด สามารถหาจุดพิกัด (X,Y,Z) นั้นได้
3. **Calculate Area** เมื่อกำหนดจุดค่าพิกัดของแปลงพื้นที่ดินรอบรูป สามารถป้อนค่าพิกัดเพื่อคำนวณหาพื้นที่ได้
4. **Mark Axis** ต้องการมาร์คแกนโดยที่รู้ค่า design และที่หน้างานทำการวางจุดด้วยกล้องโททัลสเตชั่น ทำการคำนวณด้วยเครื่องคิดเลขเพื่อหาจุดได้ 2 จุดลากเส้นเชื่อม เมื่อครบ 4 จุด จะได้เส้นตรงสองเส้นตัดกันเป็นมุม 90 องศา

วิธีการใช้งานโปรแกรม

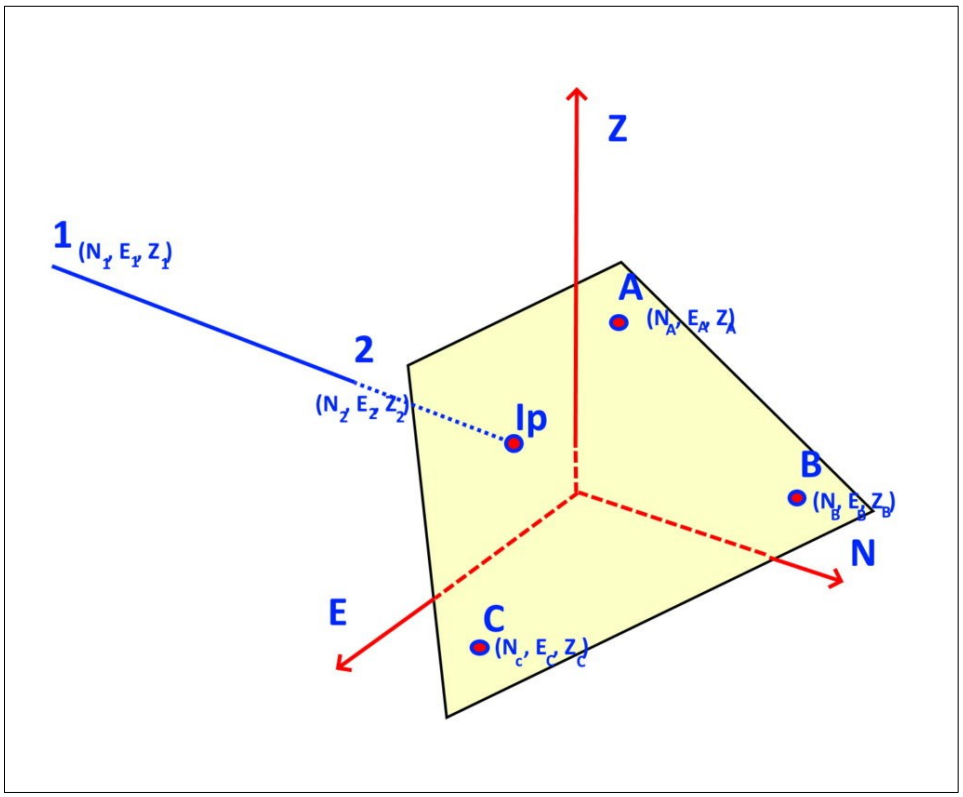
กดคีย์ “**MENU**” ที่เครื่องคิดเลข กดคีย์ลูกศรเลื่อนไปที่ไอคอน “Python” กดเข้าไป ตรงนี้ผมสร้างโฟลเดอร์ชื่อ “PROGRAM” ผมเอาซอร์สโค้ดโปรแกรมไพทอนมาไว้ที่นี่ เมื่อกด “EXE” เข้าไปจะเห็นไฟล์ซอร์สโค้ดไพทอนเรียงกันประมาณนี้ ตอนนี้จะรันไฟล์ COGOSSE4.py ซึ่งจะต้องมีไฟล์ไลบรารี pbrutils.py ด้วย



ซอร์สโค้ดไพทอนและเมนูโปรแกรมพื้นฐานงานสำรวจ ชุดที่ 4 (COGO Selected Serie 4)

Ext Line to Plane

โปรแกรมช่วยคำนวณการยืดเส้นตรงในระบบสามมิติไปตัดหรือทะลุระนาบสามมิติ โดยที่ทราบค่าพิกัดของเส้นตรง 2 จุด (X, Y, Z) และทราบค่าพิกัดของระนาบ 3 จุด (X, Y, Z)



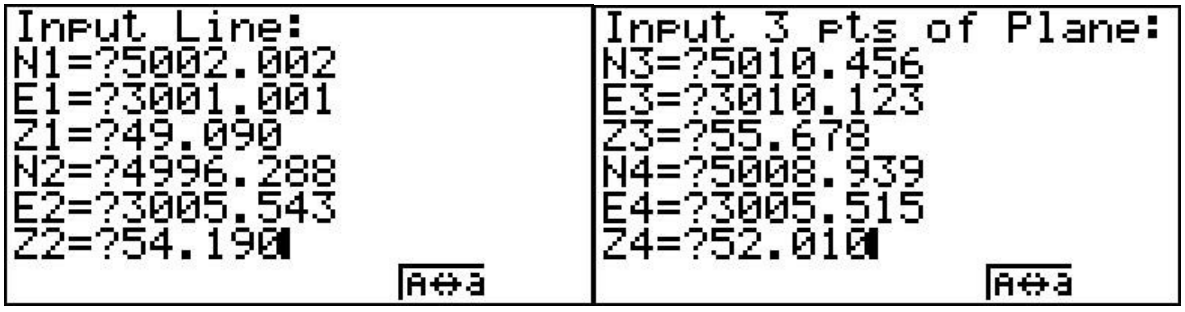
ไดอะแกรมแสดงการยืดเส้นตรงไปตัดกับระนาบในระบบพิกดสามมิติ ที่จุด IP

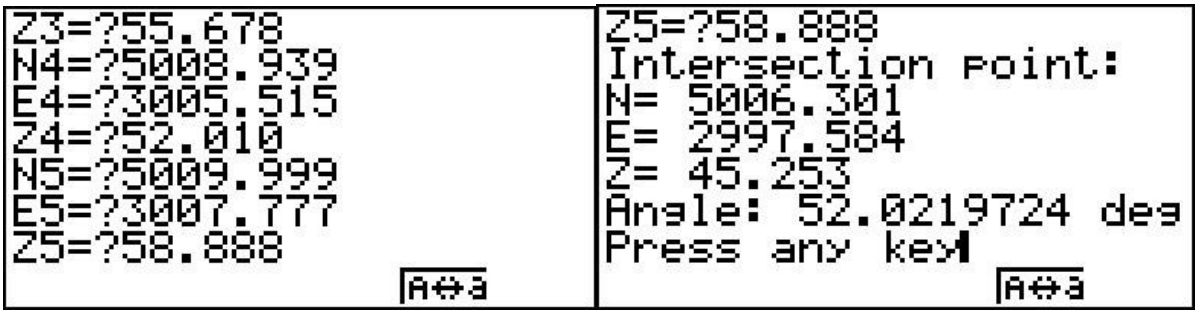
กำหนดค่าพิกัดเส้นตรงจุดที่ 1 (N=5002.002, E=3001.001, Elev=49.090) และจุดที่ 2 (N=4996.288, E=3005.543mE, Elev=54.190) และกำหนดระนาบ

1. จุดที่ 1 N=5010.456, E=3010.123, Elev=55.678 m
2. จุดที่ 2 N=5008.939, E=3005.515, Elev=52.010 m
3. จุดที่ 3 N=5009.999, E=3007.777, Elev=58.888 m

ต้องการหาจุดที่เส้นตรงไปตัดระนาบและหามุมที่ทำกับระนาบ

ที่เมนูกดคีย์ “1” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัดเส้นตรงจุดที่ 1 และจุดที่ 2 จากนั้นจะถามค่าพิกัดจุดของระนาบจำนวน 3 จุด





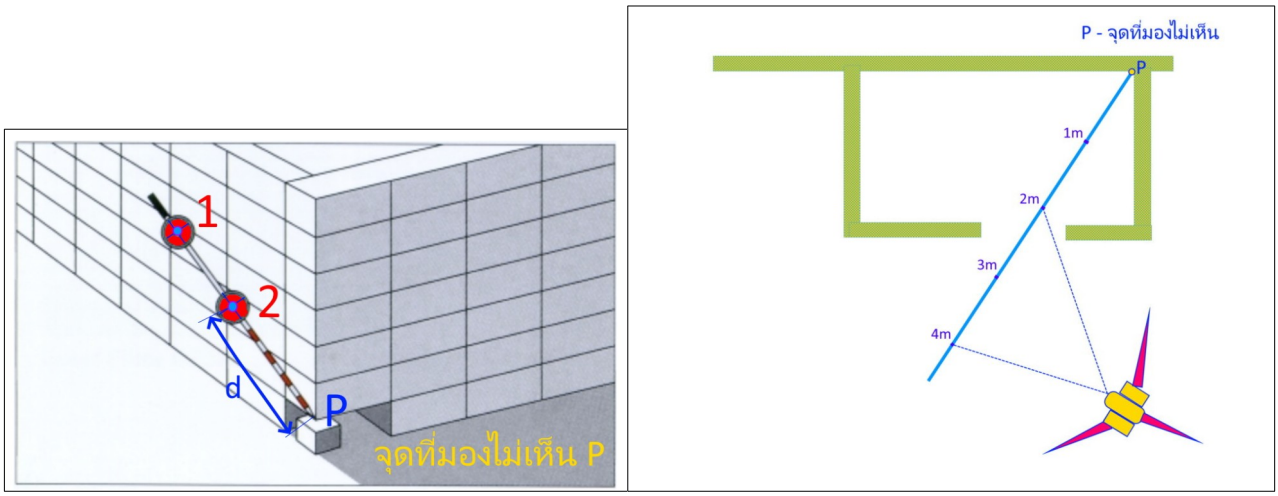
การป้อนค่าและผลลัพธ์การคำนวณ Ext Line to Plane (3D)

จะได้จุดตัด N = 5006.301, E = 2997.584 และ Z = 45.253 และมุมที่เส้นตรงทำกับ plane 52.0219724 องศา จากระนาบ X-Y

Ext Pt from 3D Line

โปรแกรมสำหรับยืดเส้นตรงในระบบ 3 มิติไปตามระยะทางที่กำหนด จากไดอะแกรมด้านล่างรูปซ้ายเป็นเป้าที่มี 2 ปริซึม กล้องโททัลสเตชัน สามารถวัดค่าพิกัดจุด 1 (N1, E1, Z1) และจุดที่ 2 (N2, E2, Z2) เมื่อทราบความสูงโพล d ก็สามารถหาค่าพิกัดของจุด P ที่ซ่อนอยู่ได้

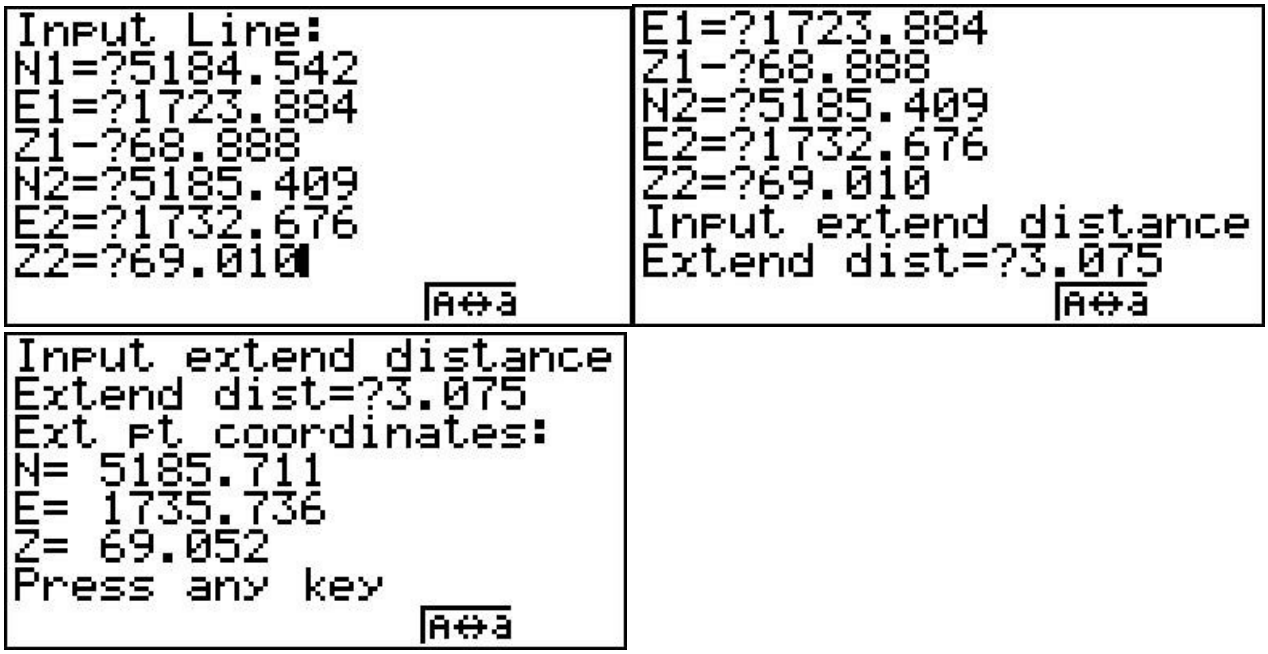
ส่วนรูปด้านขวาจะเป็นจุด P ที่มุมห้องที่กล้องโททัลสเตชันมองไม่เห็น ลากเทปเหล็กออกมาวัดค่าพิกัดแบบ Reflector-less ที่ 4 เมตร และ 2 เมตร ดังนั้นระยะทางที่ยืดไปหามุมห้องเท่ากับ 2 เมตร สามารถหาค่าพิกัดที่มุมห้องได้เช่นกัน



ไดอะแกรมแสดงการประยุกต์ใช้งาน

โจทย์กำหนดเส้นตรง จุดที่ 1 (N = 5184.542, E = 1723.884, Elev = 68.888) และจุดที่ 2 (N = 5185.409, E = 1732.676, Elev = 69.010) ยืดจุดจากจุดที่ 2 ไปอีก 3.075 เมตร ต้องการทราบค่าพิกัดของจุดนี้

ที่เมนูกดคีย์ “2” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัดเส้นตรงจุดที่ 1 และจุดที่ 2 จากนั้นจะถามระยะทางที่ต้องการยืดหรือขยายออกไป

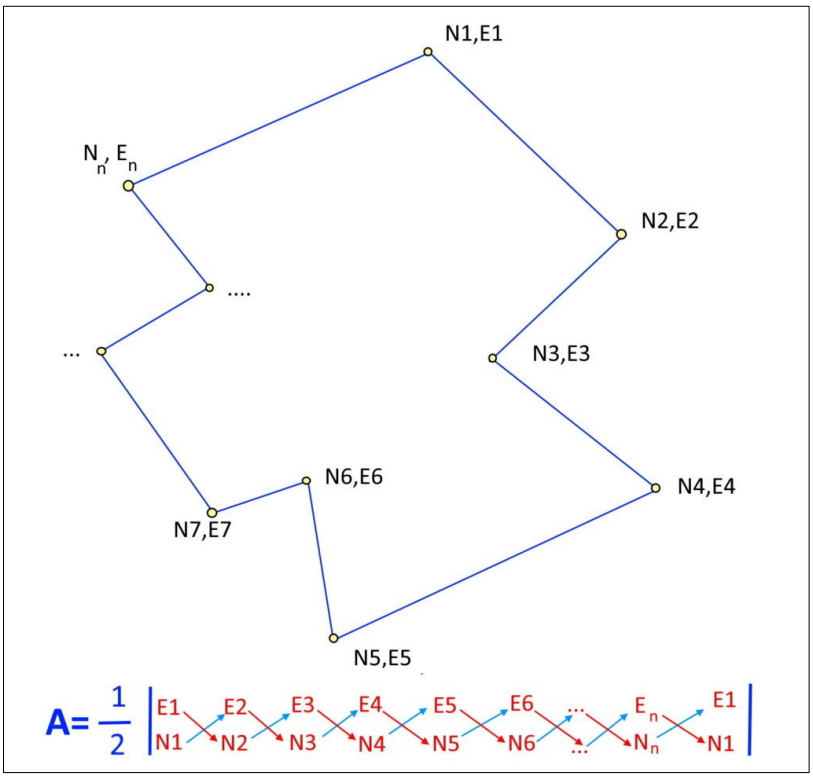


การป้อนค่าและผลลัพธ์การคำนวณ Ext Pt from 3D Line

จะได้คำตอบคือ N = 5185.711, E = 1735.736 และ Elev = 69.052

Calculate Area

การคำนวณพื้นที่เมื่อกำหนดค่าพิกัดฉากของจุดรอบแปลงปิด โดยการใช้สูตร Shoelace formula หรือสูตรผูกเชือกรองเท้า ที่อาศัยการคูณไขว้ ได้เท่าไหร่จับมาลบกันแล้วหารด้วยสอง

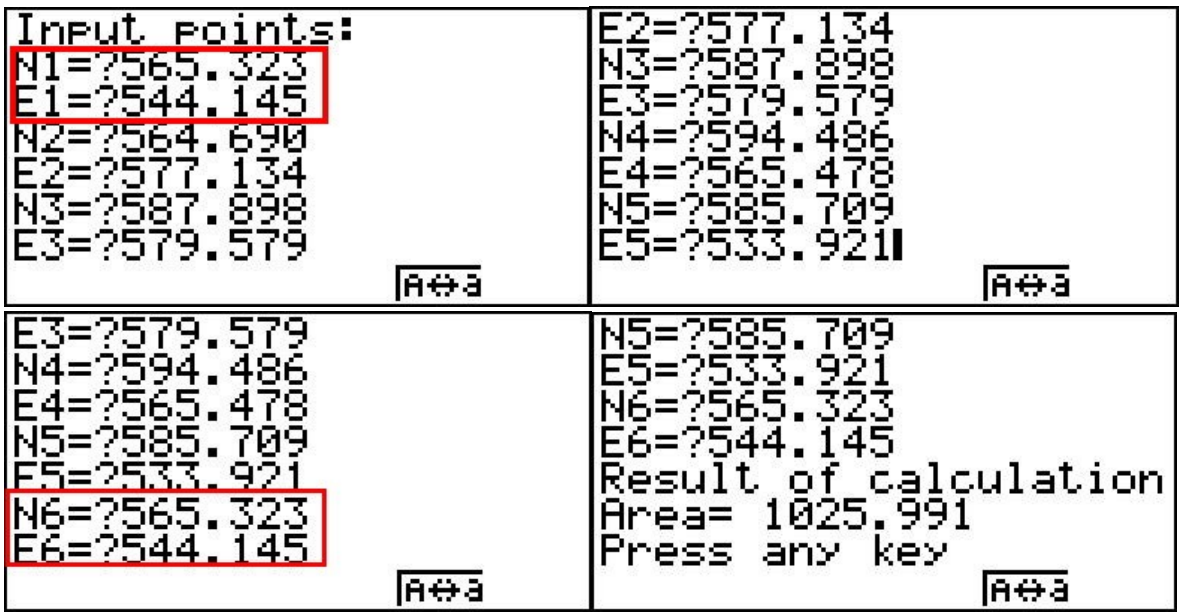


ไดอะแกรมแสดงการคำนวณพื้นที่โดยใช้สูตรคูนไขว้หรือสูตร shoelace formula

กำหนดจุดตารางด้านล่าง ต้องการหาพื้นที่

Point	Northing	Easting
1	565.323m	544.145m
2	564.690m	577.134m
3	587.898m	579.579m
4	594.486m	565.478m
5	585.709m	533.921m

ที่เมนูกดคีย์ “3” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัดจุดที่ 1 จุดที่ 2 จุดที่ 3 เมื่อต้องการปิดให้ย้อนมาป้อนจุดที่ 1 อีกครั้ง (รูปด้านล่างเน้นด้วยสี่เหลี่ยมสีแดง) โปรแกรมจะหยุดให้ป้อนค่าพิกัดและคำนวณพื้นที่มาให้



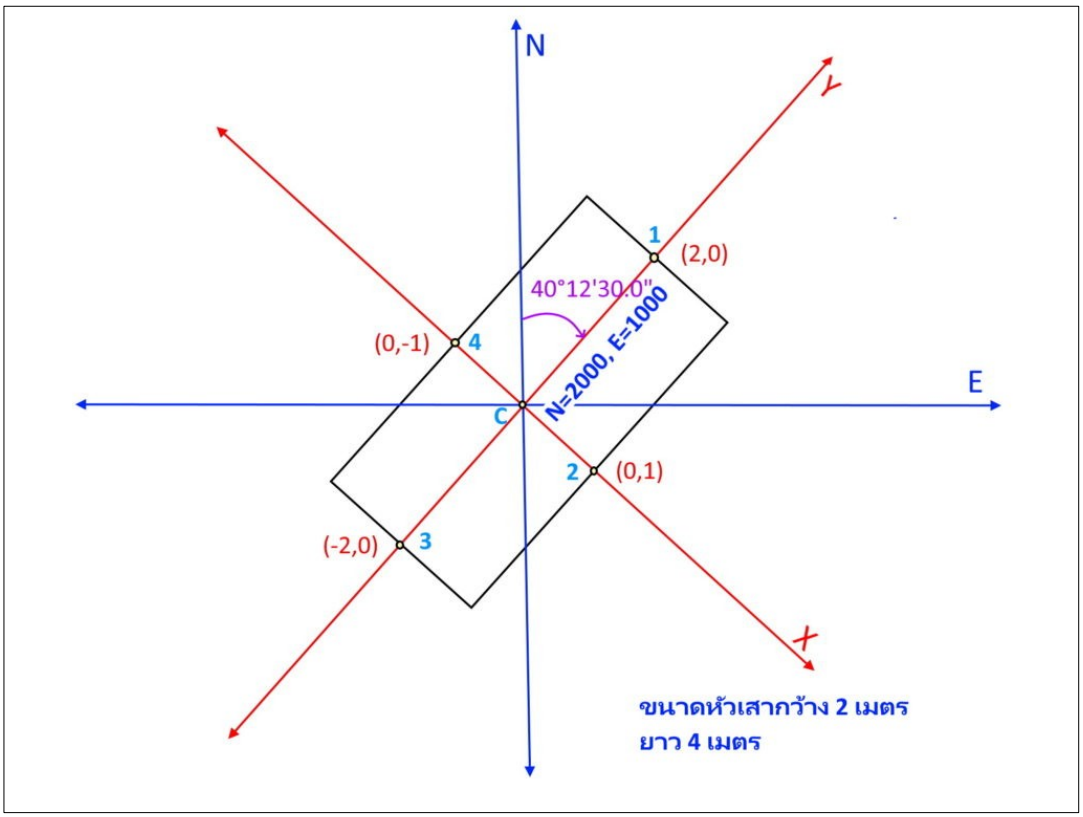
การป้อนค่าและผลลัพธ์การคำนวณหาพื้นที่ (Calculate Area)

ได้คำตอบคือพื้นที่ 1025.991 ตารางเมตร

Mark Axis

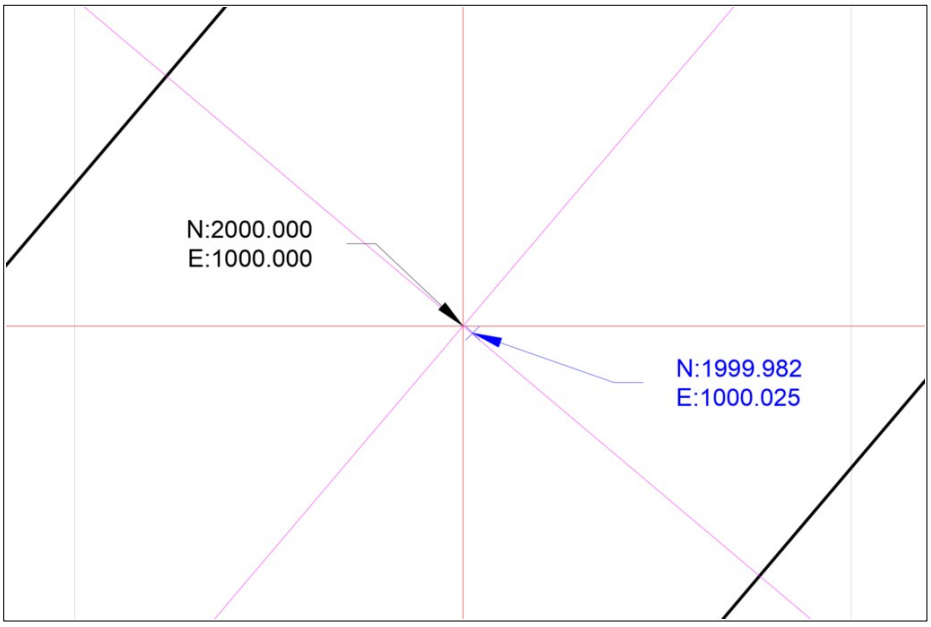
ในงานก่อสร้างบางอย่างเช่นหัวเสาหรือฟุตติ้ง จำเป็นต้องมีการตีแกนให้โดยการตีเต้า เพื่อเป็นเส้น reference ให้ทางช่างก่อสร้างหน้างานสามารถวัดออฟเซ็ทด้วยตลับเมตรหรือเรียกง่ายๆว่าแทงตลับเมตร เพื่อติดตั้งฟอร์มเวิร์คหรือทำอย่างอื่นได้ง่ายๆ สิ่งที่ช่างสำรวจต้องรู้คือค่าพิกัดจุดศูนย์กลางและอะซิมัทของแกน รูปด้านล่างเป็นหัวเสากว้าง 1 เมตร ยาว 2 เมตร จุด C หรือจุดศูนย์กลางอยู่ที่ N = 2000m, E= 1000m อะซิมัทของหัวเสาเอียงอยู่ 40°12’30” จากทิศเหนือ โจทย์ต้องการมาร์คจุดศูนย์กลาง และมาร์คจุดหมายเลข 1, 2, 3 และ 4 จากนั้นก็ตีเต้าเส้น 1-3 (แกน Y) เส้น 2-4 (แกน X)

ถ้าคิดว่าแกน YX มีจุดกำเนิด (origin) ผ่านจุดศูนย์กลางหัวเสาพอดี ค่าพิกัดจุดที่ 1 จะเป็น Y = 2, X = 0 จุดที่ 2 Y = 0, X = 1 จุดที่ 3 Y = -2, X = 0 และจุดที่ 4 Y = 0, X = -1



ที่เมนูกดคีย์ “4” และกดคีย์ “EXE” โปรแกรมจะถามค่าพิกัด (Design) จากนั้นจะถามค่าพิกัดที่เก็บได้ (Actual) จากหน้างาน ณ ตอนนั้นจากกล้องโททัลสเตชั่น จากนั้นจะคำนวณค่า Deviated เมื่อเทียบกับแกน NE และ แกน YX เราจะใช้ค่านี้ในการขยับจุดเพื่อให้เข้าหาแกน

ต้องการมาร์คจุด C หรือจุดศูนย์กลาง วัดค่าพิกัดได้ N = 1999.982 E = 1000.025

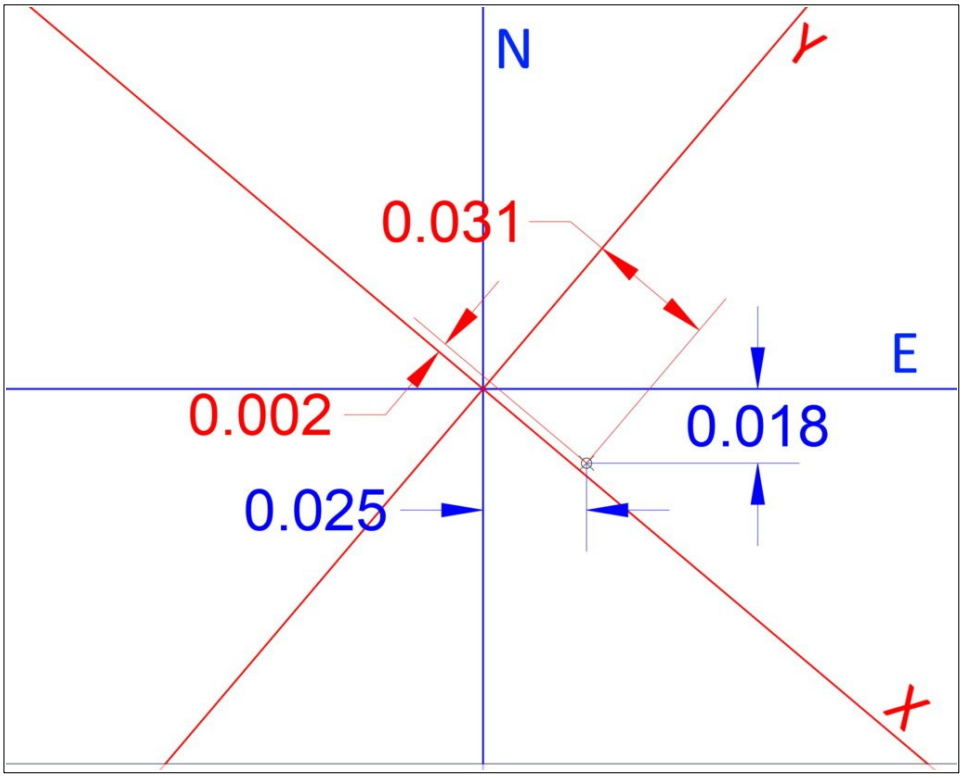


```
5:Exit
Selection[1-5]4
Mark Axis.
Input Design:
N=?2000
E=?1000
Azimuth=?40-12-30
                A⇔a
```

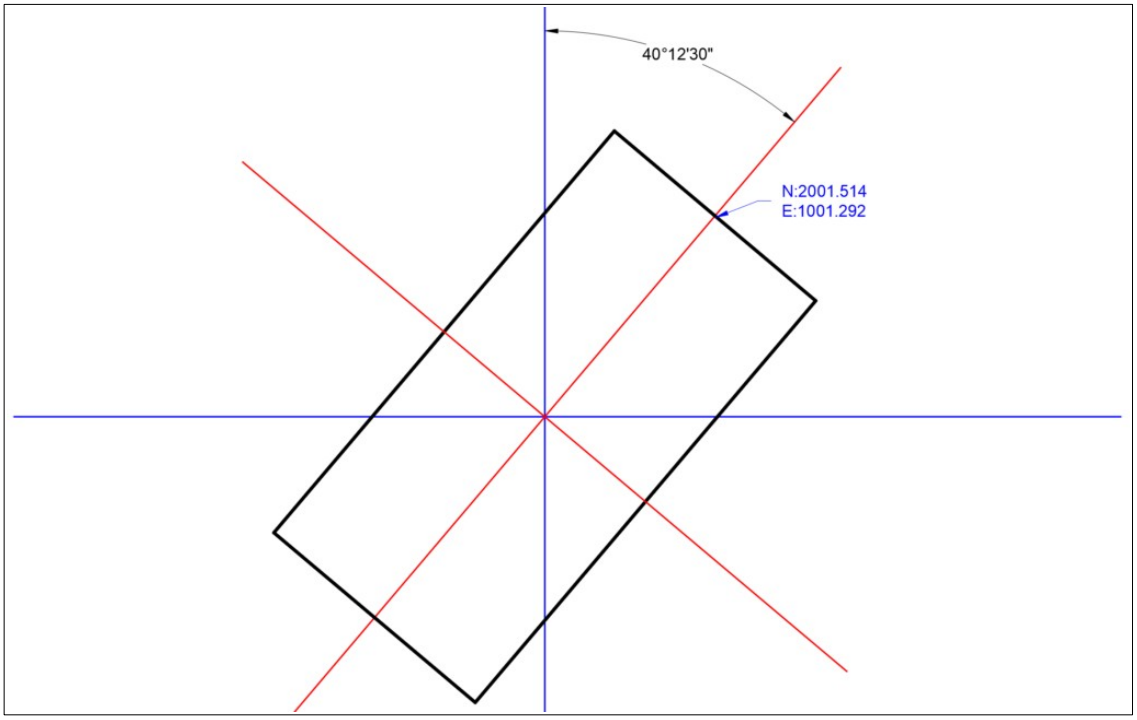
```
Input Design:
N=?2000
E=?1000
Azimuth=?40-12-30
Input Actual:
N=?1999.982
E=?1000.025
                A⇔a
```

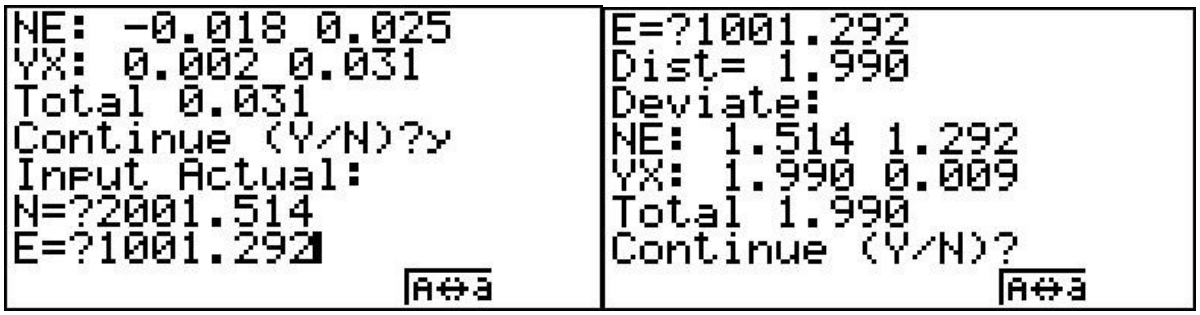
```
E=?1000.025
Dist= 0.031
Deviate:
NE: -0.018 0.025
YX: 0.002 0.031
Total 0.031
Continue (Y/N)?
                A⇔a
```

จะได้ผลลัพธ์ NE: -0.018 0.025 และ **YX: 0.002 0.031** ตัวเลขนี้บอกอะไรเราได้บ้าง ผมขยายรูปด้านบนและเขียนขนาดมิติ ค่า ΔN = -0.018 ΔE = +0.025 เมื่อเทียบกับพิกัดฉากที่เราใช้คือ NE ส่วนพิกัดฉาก YX คือพิกัดฉากของหัวเสาโดยที่ค่า **ΔY = +0.002 ΔX = +0.031** แสดงว่าจุดใหม่ที่จะวางเกินไป 0.002 ต้องขยับตามแกน Y ลงมาให้เท่ากับศูนย์ และเช่นเดียวกันเกินไป 0.031 ต้องขยับตามแกน X ไปด้านซ้ายให้เท่ากับศูนย์ ขยับจุดใหม่ป้อนค่าพิกัดเข้าโปรแกรมปรับจนได้ค่าเบี่ยงเบนเท่ากับศูนย์ YX: 0.000 0.000 จะได้จุดศูนย์กลาง



ต่อไปมาร์คจุดที่ 1 ถ้าคิดว่าทำการวางจุดอ่านจากกล้องโททัลสเตชั่นได้ N = 2001.514 E = 1001.292 ทำการป้อนค่าเข้าไปในโปรแกรม





จะได้ค่าเบี่ยงเบน NE: 1.514 1.292 และ **YX: 1.990 0.009** ในที่นี้เราต้องการค่าพิกัดเทียบกับแกน YX โดยที่ Y = 2, X = 0 (ดูรูปแรก) ดังนั้นต้องขยับไปตามแกน Y ขึ้นไปเท่ากับ 0.010 ถึงจะได้ 2 เมตรพอดี และตัวเลข 0.009 เกินจากแกน X ไปด้านขวา ต้องขยับไปด้านซ้าย 0.009 เพื่อลดลงเท่ากับศูนย์ เมื่อได้จุดที่ต้องการจะได้ค่า YX: 2.000 0.000 ทำการวางจุดที่ 2, 3 และ 4 ด้วยวิธีการเดียวกันนี้ จากนั้นทำการตีเต๊าเชื่อมจุดจะได้เส้น 2 เส้นตัดกันเป็นมุมฉาก

สรุป

ก็จบไปแล้วครับชุดโปรแกรมพื้นฐานงานสำรวจชุดที่ 4 (COGO Selected Serie 4) จะเห็นแต่ละโปรแกรมไม่มีกราฟฟิควาดรูปมาให้ดูด้วย ผมพยายามลองวาดด้วยไลบรารี matplotl.py แต่โปรแกรมหลุดบ่อย เลยล้มเลิกไป อนาคตถ้าคาสิโอเตรียมไลบรารีมาดีและเสถียรกว่านี้ ผมจะย้อนมาแก้ไขโค้ด เพิ่มส่วนวาดรูป

หลายปีที่ผ่านมายอมรับว่าไม่ได้ติดตามตลาดเครื่องคิดเลขมากนัก ผมลองดูเครื่องคิดเลขในท้องตลาดตอนนี้ยี่ห้อ TI ก็มีหลายรุ่นที่เขียนภาษาไพทอนได้ แต่ที่เหนือกว่าคาสิโอ คือมีไลบรารีต่างๆที่ TI เตรียมมาให้ด้วยเช่นฟังก์ชั่นการวาดรูป การอ่านเขียนไฟล์ แต่เมื่อข้ามไปยี่ห้อ HP รุ่น Prime พบว่าสถาปัตยกรรมเหมือนของโทรศัพท์มือถือคือใช้ซีพียูของอาร์ม (Arm V9) ซีพียูแรงกว่าแต่ต้องแลกกับพลังงานที่มากกว่า ใช้ถ่านชาร์จอาจจะต้องชาร์จกันบ่อย ที่น่าสนใจคือโปรแกรมมิ่งใช้ภาษาปาสคาล (Pascal-like) ที่ผมคุ้นเคยอยู่ ว่างๆว่าจะถอยมาเล่นๆอีกสักเครื่อง ราคาก็สูงอยู่ขนหน้าแข้งคงจะร่วงอีกหลายเส้น โปรดติดตามบทความตอนต่อไปครับ