

การควบคุมเครื่องมือวัดด้วยโปรแกรม MATLAB

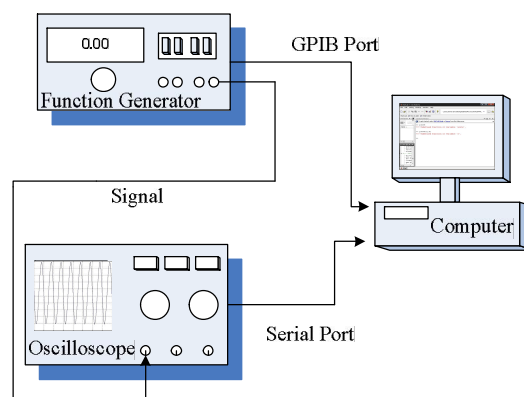
Instrument Control with MATLAB Program

นที ทองอุ่น¹ และ ประเสริฐ ลินน้อย²

1. บทนำ

การควบคุมเครื่องมือวัดผ่านคอมพิวเตอร์ PC เป็นวิธีการหนึ่งที่นิยมใช้ในกระบวนการทดสอบ ซึ่งต้องใช้เครื่องมือวัดหลายๆ ประเภทพร้อมกันในหลายสถานะการทดสอบ และโปรแกรมที่ได้รับความนิยมนำมาใช้งาน เพื่อควบคุมเครื่องมือวัดก็มีหลายโปรแกรม เช่น โปรแกรมภาษาซี โปรแกรม LabVIEW และ โปรแกรม MATLAB สำหรับโปรแกรม MATLAB เป็นโปรแกรมคำนวณทางวิทยาศาสตร์และวิศวกรรม ที่ได้รับความนิยมอย่างกว้างขวาง เนื่องจากความสามารถในการแก้ปัญหาทางคณิตศาสตร์ที่มีความซับซ้อนด้านวิศวกรรม โปรแกรม MATLAB ยังมีกล่องเครื่องมือ (Toolbox) ให้เลือกใช้ตามความต้องการในงานแต่ละประเภท เช่น Control System, Signal Processing, Image Processing เป็นต้น ส่วนงานด้านวิศวกรรมระบบเครื่องมือวัด โปรแกรม MATLAB ได้ออกแบบกล่องเครื่องมือสำหรับใช้ควบคุมเครื่องมือวัด มีชื่อว่า Instrument Control Toolbox รองรับการใช้งานร่วมกับการเชื่อมโยงแบบ GPIB, Serial, TCP/IP UDP และ VISA Standard

บทความนี้ได้นำเสนอการควบคุมเครื่องมือวัด ด้วย Instrument Control Toolbox ของโปรแกรม MATLAB ผ่านช่องสัญญาณ GPIB และ Serial ของเครื่องมือวัด 2 ประเภท คือ



รูปที่ 1 การติดต่อเครื่องมือวัดผ่าน PC

เครื่องกำเนิดสัญญาณของ Tektronix รุ่น AFG 320 และ ออสซิลอสโคปของ Tektronix รุ่น TDS 220 ส่วนโปรแกรมที่ใช้ควบคุมคือ MATLAB เวอร์ชัน 2007a ของบริษัท Mathworks

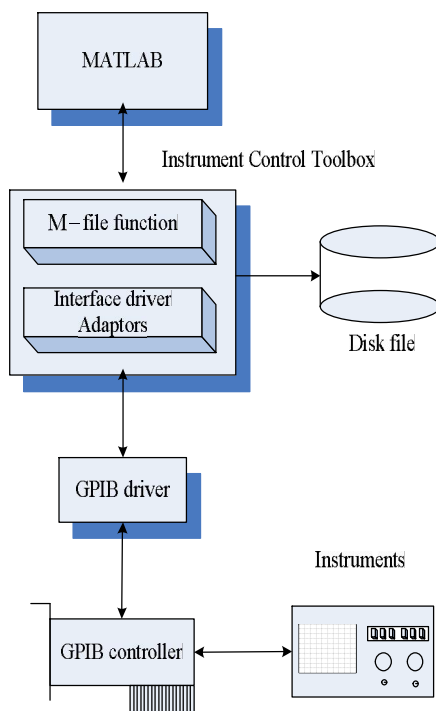
2. กล่องเครื่องมือ Instrument Control

Instrument Control Toolbox คือ กลุ่มฟังก์ชัน m-file ที่สร้างบน MATLAB ความสามารถของกล่องเครื่องมือคือ 1) รองรับการใช้งานของช่องสัญญาณแบบต่างๆ 2) มีฟังก์ชันให้ใช้งานร่วมกันระหว่างโปรแกรม MATLAB กับเครื่องมือวัดแบบต่างๆ 3) สามารถบันทึกข้อมูลและรายละเอียดของเหตุการณ์ในรูปของ Text File 4) ชุดคำสั่งมีความง่ายต่อการใช้งาน

¹ อาจารย์, ภาควิชาวิศวกรรมเครื่องมือวัดและอิเล็กทรอนิกส์, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

² เจ้าหน้าที่ห้องปฏิบัติการ, ภาควิชาวิศวกรรมไฟฟ้า, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

แสดงผลแบบกราฟฟิก กล่องเครื่องมือประกอบด้วยส่วนสำคัญ 2 ส่วน คือ m-file เป็นข้อกำหนดทางภาษา ใช้เขียนควบคุมเครื่องมือวัดบนโปรแกรม MATLAB และ Interface Driver Adaptor มีหน้าที่ส่งผ่านข้อมูลระหว่างเครื่องมือวัดกับโปรแกรม MATLAB ลักษณะการใช้ Instrument Control Toolbox ควบคุมเครื่องมือวัดมีอยู่ 2 ลักษณะ คือ แบบ Interface Object



รูปที่ 2 โค้ดโปรแกรมการส่งผ่านข้อมูล

และแบบ Device Object แบบแรกคือ การสื่อสารชนิดเชื่อมโยงโดยตรง เขียนคำสั่งผ่าน Command Line ของโปรแกรม MATLAB โดยมีข้อกำหนดภาษาตามลักษณะของผู้ผลิตเครื่องมือวัดรายนั้นๆ [1-2] สำหรับแบบที่ 2 ใช้งานง่ายกว่าแบบแรก เนื่องจากผู้ผลิตโปรแกรม MATLAB ได้ออกแบบ Instrument Drivers ซึ่งรวบรวมกลุ่มคำสั่งโปรแกรมของเครื่องมือวัดจากหลายผู้ผลิต ผู้ใช้สามารถดึงกลุ่มคำสั่งไปใช้

งานได้อย่างสะดวก แต่กรณีที่ผู้ใช้ มีเครื่องมือวัดยี่ห้อและรุ่นไม่ตรงกับกล่องเครื่องมือที่กำหนดไว้ แต่เป็นชนิดเดียวกัน ผู้ใช้สามารถแก้ไข (Edit) กลุ่มคำสั่งของ Drivers ที่มีในโปรแกรม MATLAB ให้ใช้งานได้กับเครื่องมือวัดที่มีอยู่ โดยรายละเอียดเพิ่มเติมหาได้จากเว็บไซต์ของบริษัท Mathworks [3]

3. การเขียนคำสั่งเครื่องมือวัดโดย Interface Object

จากที่กล่าวมาแล้ว วิธี Interface Object เป็นการเขียนคำสั่งโดยตรงผ่าน Command Line คือ การโปรแกรมแล้วตรวจสอบไวยากรณ์ของภาษาแต่ละบรรทัด หากผิดจะมีข้อความแจ้งเตือนทันที หลักการติดต่อของวิธีนี้ประกอบด้วย 4 ขั้นตอน คือ

3.1 การสร้าง Object

แบบนี้คือ การสร้างตัวแปรให้เก็บข้อมูลการติดต่อสื่อสาร โดยผ่านการพิมพ์ข้อความโดยตรงลงบน Command Line เช่น

```
>>g = gpib('ni',0,1)
```

คำสั่งด้านบนหมายความว่ากำหนดให้ตัวแปร g ติดต่อบน GPIB มีบอร์ดเชื่อมโยงของบริษัท National Instrument (ni) ซึ่งติดตั้งบน PC เลข 0 หมายถึง การกำหนดแอสเคอร์สให้กับบอร์ดดังกล่าวโดยเริ่มต้นที่ 0 ส่วนเลข 1 คือ การอ้างแอสเคอร์สให้เครื่องมือวัดซึ่งเชื่อมโยงกับบอร์ดหมายเลข 0 เริ่มต้นที่เลข 1 หากเป็นบอร์ดยี่ห้ออื่นให้ตรวจสอบจาก Help ว่าโปรแกรม MATLAB ให้การรองรับหรือไม่ เช่นเดียวกันหากเป็นแบบ Serial คือ

```
>>s = serial('com1')
```

คำสั่งด้านบนคือ การกำหนดให้ตัวแปร s ติดต่อบน Serial โดยกำหนดช่องทางการสื่อสารที่ช่อง com1 ของ PC

3.2 การเปิดช่องทางติดต่อ

คือ การเปิดช่องทางการติดต่อผ่านตัวแปรที่กำหนดไว้

```
>>fopen(g) ของ GPIB Port
```

```
>>fopen(s) ของ Serial Port
```

3.3 การเขียนและอ่านข้อมูล

คำสั่งเขียนข้อมูลคือ `fprintf()` และคำสั่งอ่านข้อมูลคือ `fscanf()` เช่น หากต้องการให้เครื่องกำเนิดสัญญาณจ่ายสัญญาณรูปคลื่น Ramp มีค่าขดเท่ากับ 5 โวลต์ ความถี่ 10 กิโลเฮิร์ต ที่ช่อง 1 ต้องพิมพ์คำสั่งดังต่อไปนี้

```
>>fprintf(g, 'SOURce1:FUNCTION RAMP');
```

```
>>fprintf(g, 'SOURce1:FREQuency 1E+4');
```

```
>>fprintf(g, 'SOURce1:VOLTage:AMPLitude 5');
```

ภายในวงเล็บของ `fprintf` ต้องกำหนดก่อนว่าข้อมูลที่ต้องการส่งไป เป็นของตัวแปรใดซึ่งกำลังติดต่อกับเครื่องมือวัด จากนั้นจึงพิมพ์ข้อมูลที่ต้องการลงไป จากโปรแกรมที่พิมพ์ลงไป คำว่า `SOURce` นั้นสามารถที่จะพิมพ์เพียงอักษรตัวใหญ่ `SOURCE` ได้ เครื่องกำเนิดสัญญาณสามารถเข้าใจได้แล้วว่าจะหมายถึงการกำหนดช่องสัญญาณ แต่ที่พิมพ์ให้ครบก็เพื่อความเข้าใจของผู้ใช้เอง โดยอักษรที่เหลือนั้นต้องเป็นตัวเล็ก คำสั่งอื่นก็เช่นเดียวกัน ทั้งนี้คำสั่งของโปรแกรม ต้องศึกษาจากคู่มือของผู้ผลิตเครื่องมือวัดในแต่ละราย ซึ่งในบทความนี้ใช้คู่มือของ Tektronix รุ่น AFG 320 [1] จากคำสั่งที่ผ่านมาจะสังเกตได้ว่า ต้องพิมพ์ถึง 3 บรรทัด โปรแกรม MATLAB ยึดหยุ่นให้สามารถพิมพ์ในลักษณะผสมแล้วก็ได้ผลเช่นเดียวกัน จากตัวอย่างเดิมคือ

```
>>fprintf(g, ':SOUR1:FUNC RAMP; :SOUR1:FREQ 1E+4; :SOUR1:VOLT:AMPL 5');
```

ส่วนการเขียนข้อมูล เช่น ต้องการให้ออสซิลโลสโคป อ่านค่าขดถึงขดและความถี่จากเครื่องมือวัด มีโปรแกรมดังนี้

```
>>fprintf(s, ':MEASUrement:MEAS1:TYPE PK2PK;:MEASUrement:MEAS1:VALUE?');
```

```
>>pk=fscanf(s)
```

```
>>fprintf(s, ':MEASUrement:MEAS1:TYPE FREQ; :MEASUrement:MEAS1:VALUE?');
```

```
>>f=fscanf(s)
```

ลักษณะคำสั่งที่เขียน อันดับแรกต้องกำหนดเครื่องมือวัด คือ ชนิดของค่าที่ต้องการจะวัด จากนั้นจึงอ่านค่าที่ได้จากเครื่องมือวัดมาเก็บในตัวแปร `pk` และ `f` ที่กำหนดขึ้นมา หากมีข้อความ `Time Out` ขึ้นมาหลังจากที่อ่านค่า หมายความว่าหมดเวลาจากการอ่านข้อมูลนั้นคือ เครื่องมือวัดไม่เข้าใจคำสั่งที่ต้องการให้วัด ให้ตรวจสอบการโปรแกรมดูอีกครั้งหนึ่งว่าพิมพ์ตกหล่นหรือไม่ ข้อมูลการโปรแกรมของออสซิลโลสโคป อ้างอิงจากคู่มือ [2]

3.4 ปิดช่องสัญญาณการติดต่อและคืนหน่วยความจำ

ในกรณีที่ไม่ต้องการติดต่อเครื่องมือวัดแล้ว จะต้องปิดช่องการติดต่อและคืนหน่วยความจำให้แก่ PC โดยพิมพ์คำสั่งดังต่อไปนี้

```
>>fclose(g)
```

```
>>delete(g)
```

```
>>clear g
```

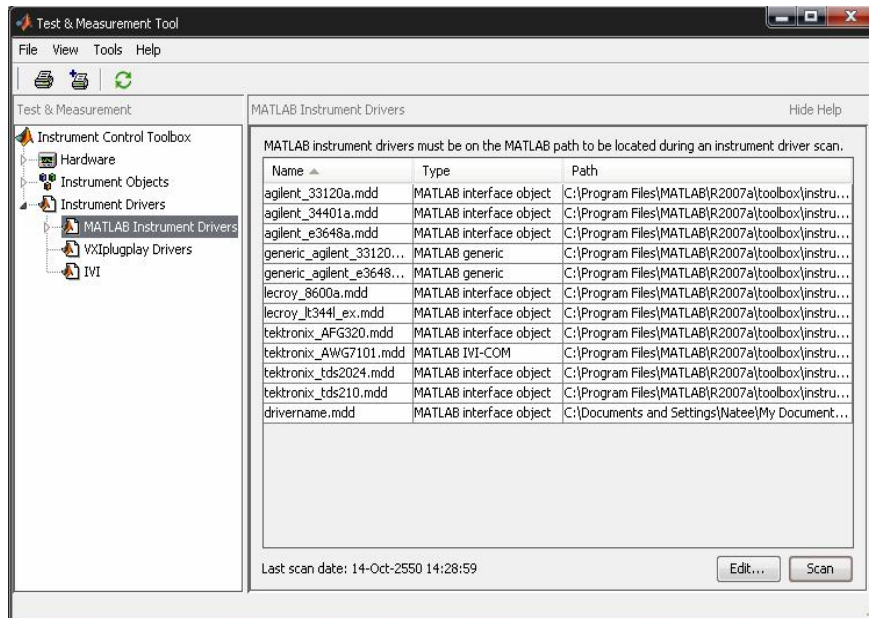
กรณีของตัวแปร `s` ให้ทำเช่นเดียวกัน

4. การโปรแกรมเครื่องมือวัดโดยวิธี Device Object

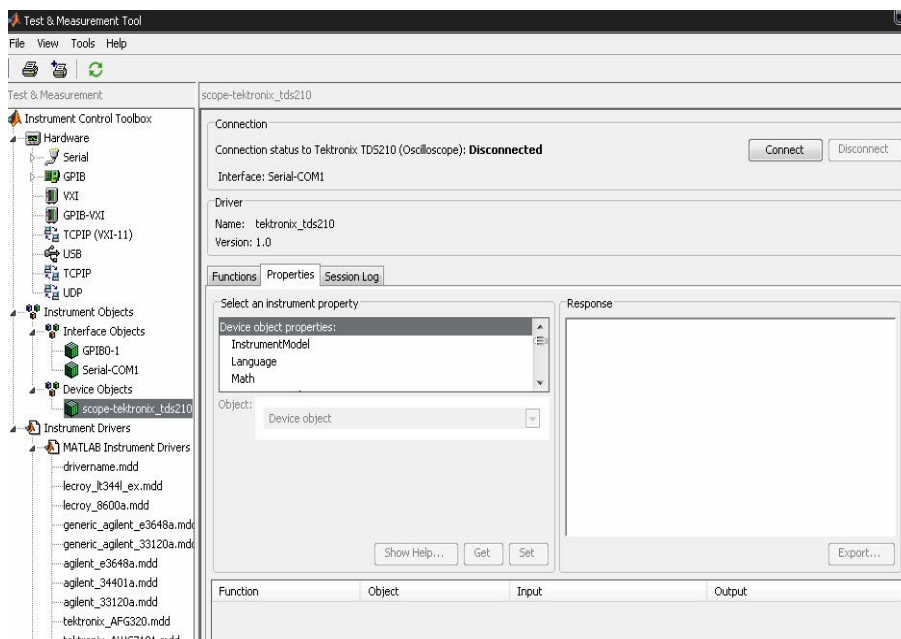
ประโยชน์ของวิธีการโปรแกรมเครื่องมือวัดโดยวิธี

Device Object เริ่มต้นจากพิมพ์คำว่า tmtool ลงบน Command

Line แล้วกด Enter จะปรากฏหน้าต่างที่เป็นรายชื่อของ Instrument Drivers ในบทความนี้จะใช้ tektronix_tds210.mdd เป็น Drivers ควบคุมออสซิลอโคป Tektronix รุ่น TDS 220 ทาง Serial Port



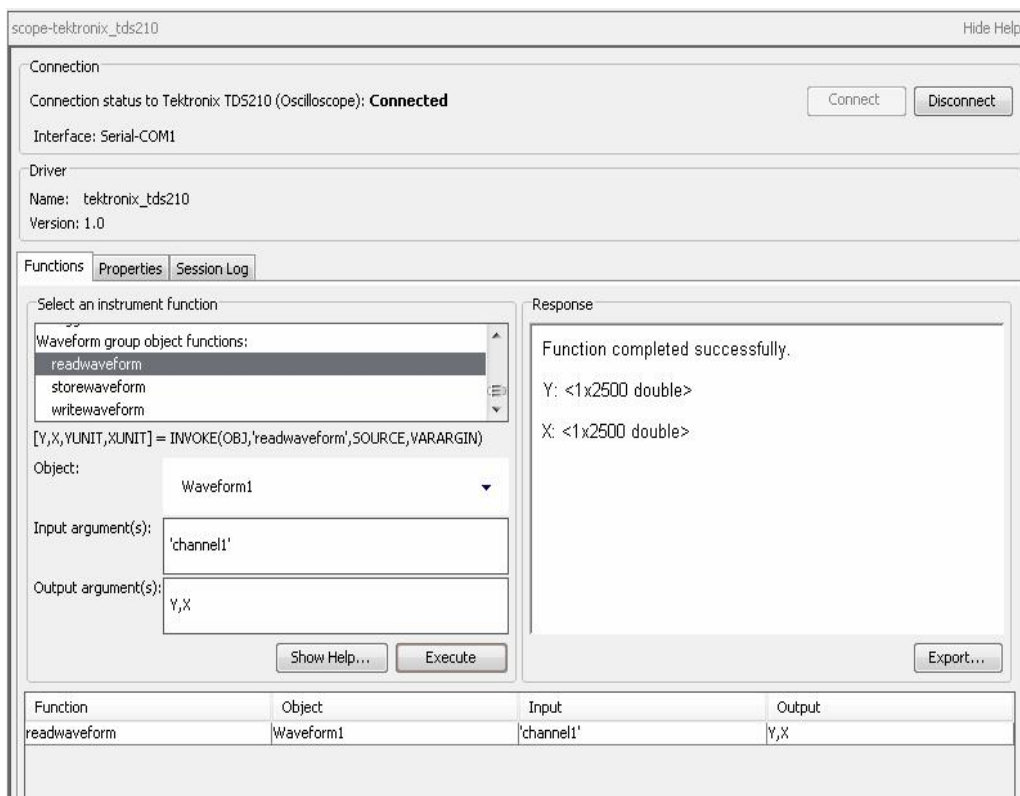
รูปที่ 3 รายชื่อ Instrument Drivers



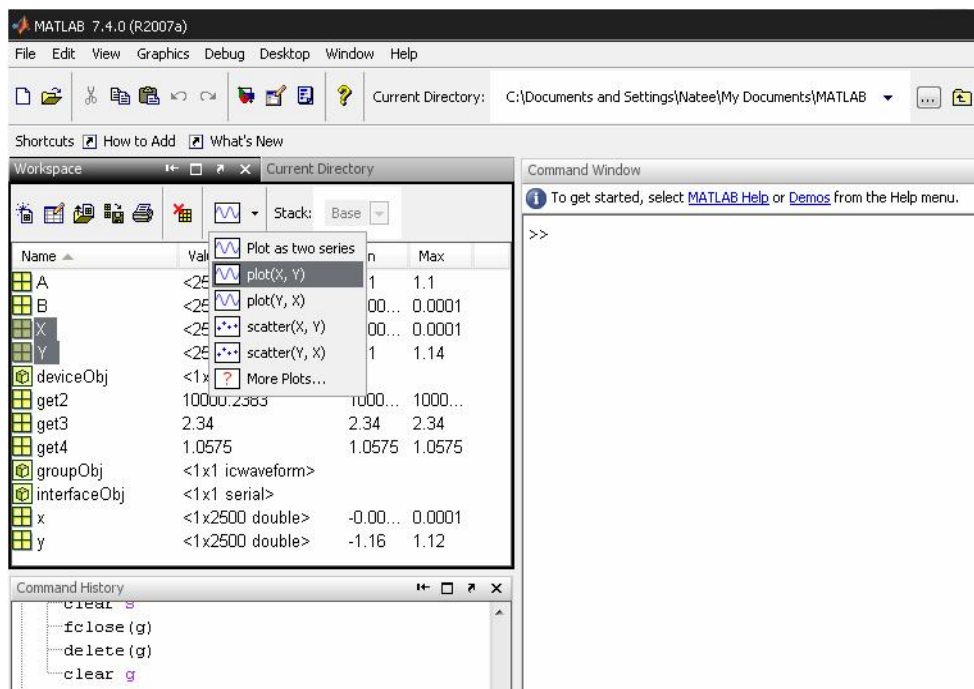
รูปที่ 4 การสร้าง Device Object ผ่าน Driver

ต่อจากนั้น สร้าง Instrument Object ชนิด Serial Port โดยคลิกที่ Instrument Object ตามด้วย Interface Object จะพบหน้าต่างขึ้นมา ให้กดปุ่ม New Object และให้เลือกชนิดการติดต่อเป็น Serial ที่ช่องทางติดต่อ com 1 ต่อไปสร้าง Device Objects ใหม่ โดยให้กดปุ่ม New Object ที่ช่อง Drivers เลือกเป็นแบบ tektronix_tds210.mdd และช่อง Interface Object ให้เลือกเป็นแบบ Serial - com 1 ในรูปที่ 4 คือ หน้าต่างใช้ติดต่อเครื่องมือวัด จากการสร้าง Device Object ในรูปพบว่า มีปุ่ม Connect ที่ใช้ติดต่อกับเครื่องมือวัดและแท็บใช้งานประกอบด้วย 3 แท็บ คือ Function, Properties และ Session log แท็บ Function คือ กลุ่มของฟังก์ชันหลักที่สามารถโปรแกรมได้ทั้งหมดกับเครื่องมือวัดที่ติดต่อกับ เช่น ต้องการนำสัญญาณที่วัดได้บนจอออสซิลโคป แสดงบนจอคอมพิวเตอร์ ให้เลือก Read

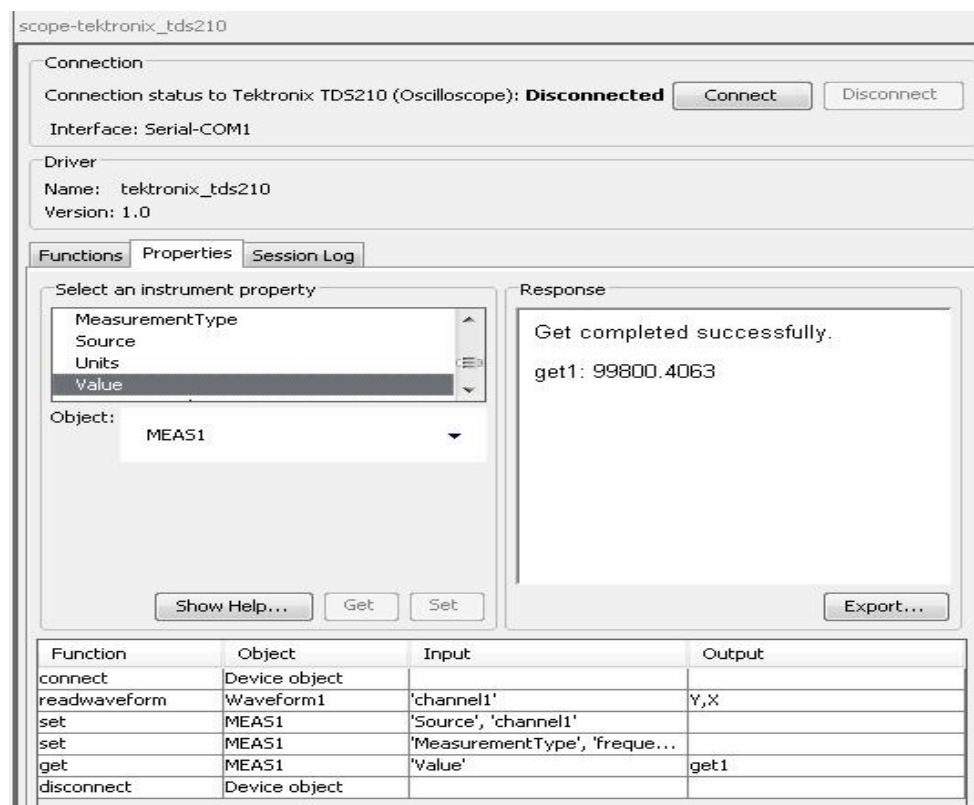
Waveform จาก Waveform Group Object Functions เมื่อเลือกไปที่รายการนี้แล้ว จะพบว่ามีช่องให้ใส่ค่า 2 ช่องคือ Input Argument(s) และ Output Argument(s) ช่อง Input ให้ใส่ 'channel1' เป็นการกำหนดว่ารับค่ามาจากช่องที่ 1 ของออสซิลโคป สำหรับช่อง Output ให้ใส่ตัวแปรที่ต้องการเก็บค่ากำหนดให้เป็น y,x จากนั้นกดปุ่ม Execute ตามรูปที่ 5 เมื่อกำหนดตัวแปร y และ x ให้เก็บข้อมูลภาพแล้ว ให้กดปุ่ม Export มุมขวาล่างสุดด้านล่าง จะปรากฏหน้าต่างถามว่าต้องการจะส่งข้อมูลไปเก็บไว้ที่ใด ให้เลือก MATLAB Workspace ในช่อง Data Destination และกดปุ่ม Export ต่อไปให้กลับไปยังหน้าต่างของ MATLAB แล้วเลือกตัวแปรที่ได้ส่งไปแล้วบน MATLAB Workspace และเลือก Plot สัญญาณแบบ x และ y ดังรูปที่ 6



รูปที่ 5 กำหนดรูปคลื่นแสดงผลบนหน้าจอคอมพิวเตอร์

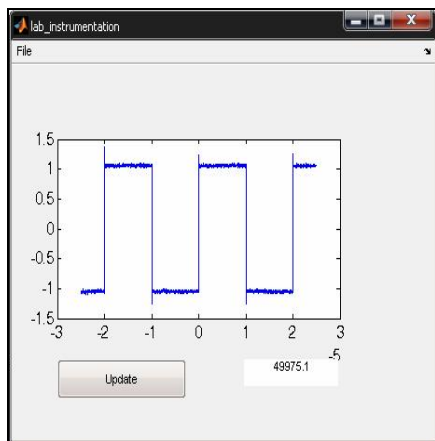


รูปที่ 6 การนำข้อมูลที่อ่านได้มาแสดงผล



รูปที่ 7 กำหนดให้เครื่องมือวัดค่าความถี่ช่อง 1

ในแท็บที่ 2 คือ Properties ภายในโหนดนี้คือ กลุ่มฟังก์ชันซึ่งเป็นคุณสมบัติย่อยของออสซิลโลสโคป ที่ผู้ใช้สามารถเลือกกำหนดได้โดยตรง เช่น ต้องการให้เครื่องมือวัดความถี่ที่ช่องที่ 1 ดังรูปที่ 7 แท็บสุดท้ายคือ Session Log เป็นการแสดงโปรแกรมทั้งหมดจากที่ได้เลือกไว้ใน 2 แท็บ แรก ในรูปของ m-file เพื่อนำไปประยุกต์ใช้งานร่วมกับ User Interface ของ MATLAB หรือที่รู้จักกันในชื่อว่า GUI จากตัวอย่างที่ได้กำหนดคุณสมบัติของเครื่องมือวัดไว้แล้วในข้างต้น ให้นำไปประยุกต์เข้ากับ GUI ได้ผลลัพธ์ดังรูปที่ 8 เป็นการแสดงผลการอัปเดตสัญญาณและจะมีช่องแสดงค่าความถี่ของสัญญาณ

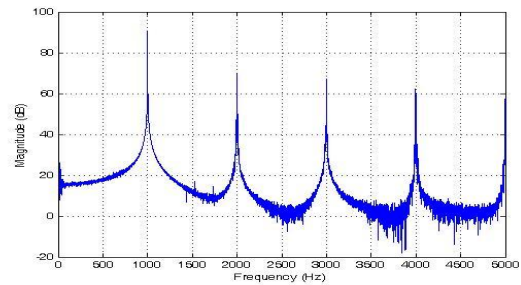


รูปที่ 8 User Interface ของสัญญาณที่วัดได้

นอกจากนี้แล้วออสซิลโลสโคปในรุ่น TDS 1000 ยังมีฟังก์ชันทางคณิตศาสตร์ให้เลือกใช้งานเช่น Fast Fourier Transform เพื่อนำสัญญาณดังกล่าวมาวิเคราะห์เชิงความถี่เช่น คำสั่งดังนี้

```
>>fprinf(s,'MATH:DEFINE "FFT(CH1,HANNING)");
```

คำสั่งดังกล่าวเป็นคำสั่งที่กำหนดให้ออสซิลโลสโคปแสดงค่าของสัญญาณที่ช่อง 1 ในรูปของโดเมนความถี่หากต้องการนำภาพมาแสดงผลบนจอคอมพิวเตอร์ที่ติดต่อก็คือสามารถใช้คำสั่ง Hard Copy และการปรับแต่งภาพ [2-3] จะได้ลักษณะของผลลัพธ์โดเมนความถี่ดังรูปที่ 9



รูปที่ 9 การแสดงผลในโดเมนความถี่

5. สรุป

การควบคุมเครื่องมือวัดโดยโปรแกรม MATLAB สามารถใช้งานได้ 2 วิธีคือ แบบ Interface Object ใช้เขียนบน Command Line เพื่อเขียนคำสั่งโดยตรงกับเครื่องมือวัด วิธีที่ 2 คือ แบบ Device Object เป็นการโปรแกรมที่สะดวกกว่าวิธีแรก เพราะสามารถเลือกกลุ่มฟังก์ชันที่ต้องการโปรแกรมได้ทันที โดยไม่ต้องเสียเวลาพิมพ์แต่มีข้อแม้ว่า เครื่องมือวัดชนิดหรือรุ่นนั้นๆต้องมี Instrument Drivers ที่รองรับจากบริษัทผู้ผลิตโปรแกรม MATLAB หากเครื่องมือวัดไม่อยู่ในบัญชี Instrument Drivers ของโปรแกรม MATLAB ผู้ใช้ต้องแก้ไขตัว Instrument Drivers เติมขึ้นมาใช้งานเอง การควบคุมเครื่องมือวัดดังกล่าว สามารถนำไปประยุกต์ใช้งานกับ GUI ของโปรแกรม MATLAB เพื่อสร้าง User Interface ได้

เอกสารอ้างอิง

- 1 บริษัท Sony Tektronix, คู่มือการเขียนโปรแกรมของ Function Generator รุ่น AFG 320.
- 2 บริษัท Tektronix, คู่มือการเขียนโปรแกรมของ Digital Oscilloscope รุ่น TDS 220,TDS1000
- 3 <http://www.mathworks.com/>
- 4 S. Gupta and J. John, Virtual, Instrumentation Using Lab VIEW, Tata McGraw – Hill, 2005.

