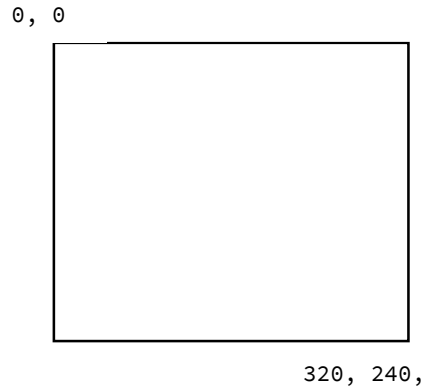


การแสดงผลผ่านจอ LCD

การใช้งาน การแสดงผลผ่าน LCD ใช้ ไลบรารี "M5Stack.h" เพื่อเรียกฟังก์ชันต่างๆ ที่ทำงานกับจอภาพ LCD และเมื่อเริ่มต้นใช้งานไลบรารีนี้จะต้อง เรียกฟังก์ชัน `begin()` ในส่วน `setup()` ก่อน นอกจากนี้ ในฟังก์ชัน `loop()` ให้ใช้ฟังก์ชัน `update()` เพื่อการปรับปรุงการทำงานของแต่ละครั้งของการทำงาน แต่ถ้าไม่มีอะไรเปลี่ยนแปลง ก็ไม่จำเป็นต้องเรียกฟังก์ชัน `update()` ก็ได้



รูป 1 ขนาดหน้าจอ LCD ของ M5Stack

ขนาดหน้าจอ LCD มีขนาด 320×240 โดยจุดเริ่มต้น อยู่ซ้ายสุด-ขวาสุด ตำแหน่งนี้คือ (0, 0) ความกว้าง 320 นั้นเรียกได้ด้วยฟังก์ชัน `width()` และ ความสูงแทนด้วยฟังก์ชัน `height()` มีค่า 240

ตัวอย่างต่อไปนี้จะแสดงการเริ่มต้นวาดเส้น แนวทแยง จาก จุดเริ่มต้น (0, 0) ไป จุดสุดท้าย (320, 240) ด้วยการเรียกฟังก์ชัน `drawLine()`

ตัวอย่าง 1

```
#include <M5Stack.h>
void setup() {
  M5.begin();
}
void loop() {
  M5.Lcd.drawLine(0,0,320,240,WHITE);
}
```

ฟังก์ชันของ M5Stack

วาดเส้น

จุดสี

การสร้างจุดสี (pixel) ใช้ `drawPixel(x0, y0, COLOR)`

เส้นตรง

โดยให้ `x0, y0` แทน จุดเริ่มต้น มี `x1, y1` แทนจุดสิ้นสุดของเส้นตรง และ `COLOR` แทนสี

`drawLine(x0, y0, x1, y1, COLOR)`

เส้นตรงแนวตั้ง

โดยให้ x0, y0 แทน จุดเริ่มต้น มี height แทนความสูง และ COLOR แทนสี

```
drawFastVLine(x0, y0, height, COLOR)
```

เส้นตรงแนวนอน

โดยให้ x0, y0 แทน จุดเริ่มต้น มี height แทนความสูง และ COLOR แทนสี

```
drawFastHLine(x0, y0, lenght, COLOR)
```

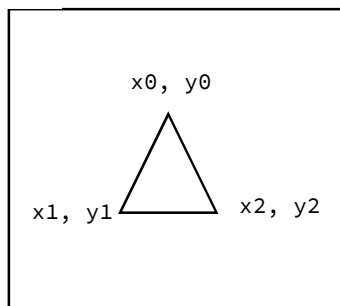
สามเหลี่ยม

ใช้การวาด (draw) ใช้ จัดสามจุดของเหลี่ยมเหลี่ยม (x, y) ซึ่งจะได้กรอบ และ fill จะเติมสี (COLOR) ในกรอบ

```
drawTrainggle(x0, y0, x1, y1, x2, y2, COLOR)
```

```
fillTrainggle(x0, y0, x1, y1, x2, y2, COLOR)
```

0, 0



320, 240

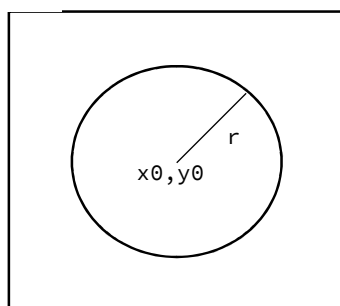
วงกลม

ใช้ศูนย์กลาง (x0, y0) และ r แทนรัศมี

```
drawCircle(x0, y0, r, COLOR)
```

```
fillCircle(x0, y0, r, COLOR)
```

0, 0



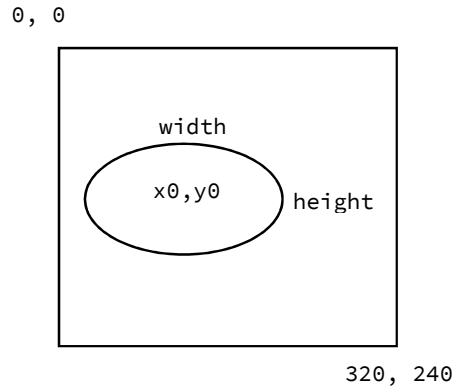
320, 240

วงรี

ใช้ศูนย์กลาง (x0, y0) และใช้ width กับ height แทนความกว้าง

```
drawCircle(x0, y0, width, height, COLOR)
```

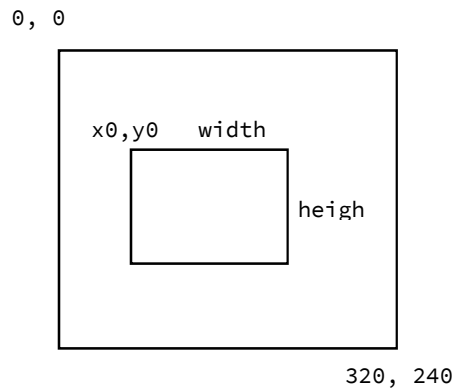
```
fillCircle(x0, y0, width, height, COLOR)
```



สี่เหลี่ยม

ใช้ จัดเริ่มต้น (x0,y0) และใช้ width กับ height แทนความกว้าง

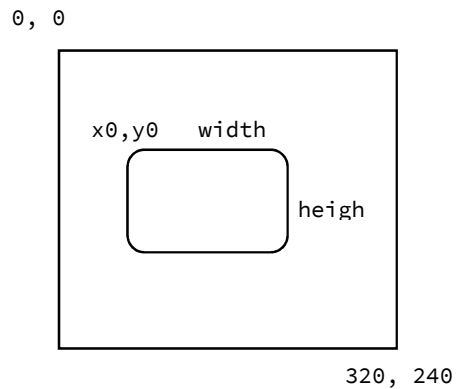
```
drawRect( x0, y0, width, height, COLOR)
fillCircle(x0, y0, width, height, COLOR)
```



สี่เหลี่ยมขอบมน

ฟังก์ชันคล้ายการวาดรูปสี่เหลี่ยมแต่เพิ่มตัวแปร radius แทนตัวเลข ความโค้งมนของมุมสี่เหลี่ยม

```
drawRoundRect( x0, y0, width, height, radius, COLOR)
fillRoundCircle(x0, y0, width, height, radius, COLOR)
```



แท่งวัดความกว้างหน้า (bar)

```
progressBar(x0, y0, width, height, progress_0_to_100)
```

การใช้งานฟังก์ชันนี้จะต้องใช้ร่วมกับ fillRect() เพื่อติกรอบนอก เช่น การถัดต้องการใช้ แท่งวัดความกว้างหน้าyak 240 มีก้าวน้ำ 20 จะใช้งานดังนี้

```
M5.Lcd.fillRect(0,0,240,20,0);  
M5.Lcd.progressBar(0,0,240,20, 20);
```

ข้อความ

กำหนดให้ (x0, y0) เป็นจุดเริ่มต้นที่วาดอักษร

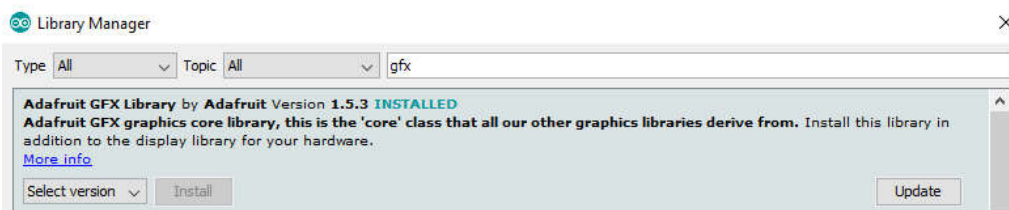
```
print("string")  
setCursor(x0, y0) แทนจุดเริ่มต้นที่จะมีการพิมพ์ จากการใช้ฟังก์ชัน print( )  
setTextColor(backgroundColor, backgroundColor)  
drawChar(x0, y0, 'A', drawingColor, backgroundColor, size )  
drawString("data", x0, y0, font)  
drawString("data", x0, y0)
```

สำหรับการวาดอักษร (drawString) สามารถเลือก ชนิดอักษร (font) ได้ ซึ่งในตัวอย่างของ M5stack มี Free_Fonts.h ซึ่งหากต้องการใช้ จะต้องนำไฟล์ Free_Fonts.h มาเก็บในตำแหน่งเดียวกับ ไฟล์ Arduino หรือใช้เมนู Sketch > Add File... ฟังก์ชันเกี่ยวกับ ชนิดอักษร มีดังนี้

```
setTextSize(1) โดย 1 เป็นค่าปริยาย มีขนาด 10 จุดสี่ การเพิ่มเป็น 2 ก็จะมีขนาด 2 เท่า หรือ 20 จุดสี่  
setTextDatum(MC_DATUM) เป็นการกำหนดให้อักษรตรงกลางหน้าจอ  
setFreeFont(FF18) กำหนดชนิดฟอนต์เป็น FF18 ซึ่งเป็นชื่อฟอนต์ฟรีของ GFXFF ตัวหนึ่ง  
drawString(text, 160, 120, GFXFF) เป็นวาดอักษรตระกูล GFXFF มีข้อความตามตัวแปร text
```

สำหรับ Free_fonts.h มี 48 ตั้งแต่ FF1 - FF48 แบบ (ชื่อฟอนต์นี้เป็นที่อยู่ (address) ของฟอนต์ มีขนาด 9, 12, 18, 24 เช่น FF1 มีขนาด 9 FF2 มีขนาด 12 FF3 มีขนาด 18 และ FF4 มีขนาด 24 ซึ่งเป็นขนาดสูงสุด

Free_fonts.h ใช้ โลบารี่ของ Adafruit GFX จึงต้องติดตั้งโลบารี่นี้ก่อน ตามหน้าต่าง Library Manager



รูป 2 Adafruit GFX Library

ไฟล์ free_fonts.h ดาวน์โหลดที่

https://github.com/Bodmer/TFT_HX8357_Due/blob/master/examples/Free_Font_Demo/Free_Fonts.h

และตัวอย่างการใช้งาน ดูที่

https://github.com/Bodmer/TFT_HX8357_Due/tree/master/examples/Free_Font_Demo

จัดเรียงฟอนต์ ดูที่

https://github.com/Bodmer/TFT_ILI9341/blob/master/examples/TFT_String_Align/TFT_String_Align.ino

ความสว่าง

ในการวัดความสว่างใช้ฟังก์ชัน `setBrightness( 0_to_255 )` โดยสว่างมากที่สุดอยู่ที่ 255

หลับและตื่น

ฟังก์ชัน `sleep( )` ใช้เพื่อลดการใช้พลังงาน หรืออยู่ในโหมด หลับ(sleep) เมื่อต้องการปลุกระบบขึ้นมาใหม่ ใช้ฟังก์ชัน `wakeup( )` นอกจากนี้ควร กำหนดความสว่าง ให้มืดที่สุด ในกรณีใช้โหมดหลับ หรือสว่างขึ้นตามความต้องการ เมื่อใช้ปลุกให้ระบบทำงาน

อื่นๆ

ยังมีฟังก์ชันอื่น ๆ อีกมาก เช่น การหมุนด้วยฟังก์ชัน `setRotation( 90 )` แทนการหมุน 90 องศา ฟังก์ชัน `invertDisplay(true)` แทนการแปลงสี จาก ด้านบวกไปด้านลบ

ตาราง 1 ข้อกำหนดค่าคงที่ของสี

Definition	Value	R	G	B
TFT_BLACK	0x0000	0	0	0
TFT_NAVY	0x000F	0	0	128
TFT_DARKGREEN	0x03E0	0	128	0
TFT_MAROON	0x7800	128	0	0
TFT_PURPLE	0x780F	128	0	128
TFT_OLIVE	0x7BE0	128	128	0
TFT_LIGHTGREY	0xC618	192	192	192
TFT_DARKGREY	0x7BEF	128	128	128
TFT_BLUE	0x001F	0	0	255
TFT_GREENYELLOW	0xB7E0	180	255	0
TFT_GREEN	0x07E0	0	255	0
TFT_YELLOW	0xFFE0	255	255	0
TFT_ORANGE	0xFDA0	255	180	0
TFT_PINK	0xFC9F	255	255	16
TFT_CYAN	0x07FF	0	255	255
TFT_DARKCYAN	0x03EF	0	128	128
TFT_RED	0xF800	255	0	0
TFT_MAGENTA	0xF81F	255	0	255
TFT_WHITE	0xFFFF	255	255	255

ตัวอย่าง 2 วาดเส้น แนวตั้งและแนวนอน

```
#include <M5Stack.h>
```

```

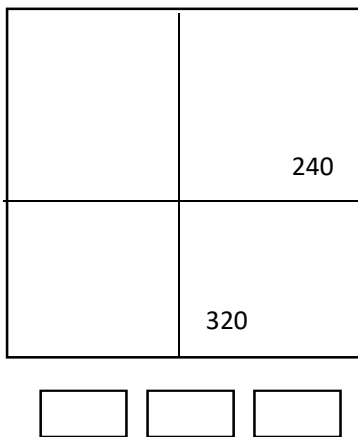
int w = M5.Lcd.width(); //LCD Width = 240
int h = M5.Lcd.height(); //LCD Height = 320

void setup() {
  M5.begin();
  M5.Lcd.drawString(String(w), 300, h/2);
  M5.Lcd.drawFastHLine(0, h/2, h, RED);

  M5.Lcd.drawString(String(h), h/2, 230);
  M5.Lcd.drawFastVLine(h/2, 0, w, RED);
}

void loop() {
}

```



ในตัวอย่างนี้ ใช้ ตัวแปร w แทน ความกว้าง ซึ่งคือความสูง ถ้าใช้ ปุ่ม 3 ปุ่ม เป็นฐาน ซึ่งพิมพ์ค่า 240 และในทางกลับกัน ใช้ h แทนความสูง ซึ่งคือความกว้าง ถ้าใช้ ปุ่ม 3 ปุ่มเป็น

ตัวอย่าง 3 การสร้างตัวเลข วิ่ง จาก 0 ถึง 9

```

#include <M5Stack.h>
int w = M5.Lcd.width(); //LCD Width
int h = M5.Lcd.height(); //LCD Height
String digitStr = "0123456789"; //digits in a string
void setup() {
  M5.begin();
}

void loop() {
  for(int i=0; i<digitStr.length(); i++){
    M5.Lcd.drawChar(w/2, h/2, digitStr[i], TFT_RED, TFT_BLACK, 3);
    sleep(1); // one second
  }
}

```

ตัวอย่าง 4 การสร้างตัวเลข วิ่ง จาก 1 ถึง 10

```
#include <M5Stack.h>
#include "Free_Fonts.h"

void setup() {
  M5.begin();
}

void loop() {
  // Set text datum to middle centre
  M5.Lcd.setTextDatum(MC_DATUM);

  // Set text colour to orange with black background
  M5.Lcd.setTextColor(TFT_WHITE, TFT_BLACK);
  M5.Lcd.setTextSize(2);

  for(int i=1; i<11;i++){
    String text = String(i);
    M5.Lcd.fillScreen(TFT_BLACK);           // Clear screen
    M5.Lcd.setFreeFont(FF18);              // Select the font
    M5.Lcd.drawString(sFF1, 160, 60, GFXFF); // Print the string name of the
font
    M5.Lcd.setFreeFont(FF16);              // Select the font
    M5.Lcd.drawString(text, 160, 120, GFXFF); // Print the string name of
the font
    delay(1000); //delay 1 second
  }
}
```

ทดลองด้วยตนเอง

1. ให้วาด รูปสี่เหลี่ยมแทน เส้น (เป็นเส้นที่มีความหนากว่าเป็นเส้น)
2. ใส่เลขตัวนับบิ่งจาก 1 - 10 บิ่งวนซ้ำไปมา ในช่องแรก (บน) ช่องล่างซ้ายแรก ในอักษร บิ่ง จาก A - F วนซ้ำไปมา
3. ใส่สี ช่อง บนเป็นพื้น ฟ้ำ อักษร เป็นสี แดง
4. วาดรูป วงกลมในขอบช่องที่สองล่างขวา
ทั้งหมดมีผลลัพธ์ดังรูปต่อไปนี้

