

시리얼 칼라 디스플레이 모듈
Serial Color Display Module
M DisplayTM

MANUAL VERSION: 201408



* MDisplay 는<UTF-8 유니코드>를 지원합니다. KS 완성형 한글코드는 사용할 수 없습니다.

*한글이 제대로 표시되지 않는다면, UTF-8 유니코드로 보내고 있는지 확인해보시기 바랍니다.

COMFILE
TECHNOLOGY

컴파일 테크놀로지(주)

등록상표

WINDOWS 는 Microsoft Corporation 의 등록상표입니다.
CUBLOC 은 Comfile Technology 의 등록상표입니다.
기타 다른 상표는 해당회사의 등록상표입니다.

알림

본 설명서의 내용은 사전 통보 없이 변경될 수 있습니다. 본 제품의 기능은 성능 개선을 위하여 사전 통보 없이 변경될 수 있습니다. 본 제품을 이 자료에서 설명한 용도 외에서 사용할 경우, 폐사에서는 어떠한 책임도 지지 않으므로 주의하시기 바랍니다. 본 제품은 컴파일 테크놀로지의 고유 기술을 사용하여 개발된 제품으로 저작권법에 의한 보호를 받고 있습니다. 따라서 본 제품 (제품에 대한 아이디어 및 설명서 및 기타 포함)의 어떠한 부분도 사전에 폐사와의 문서 동의 없이 복사되거나 변경, 재 생산할 수 없으며 또한 다른 언어로도 번역될 수 없습니다.

주의사항

인쇄된 설명서는 인쇄된 시점에서는 최신 버전이지만, 인쇄된 후 시간이 경과된 뒤에 새로운 내용이 추가되거나, 기존 내용이 바뀔 가능성이 있습니다. 최신 버전의 설명서는 항상 인터넷 홈페이지 (www.comfile.co.kr)에서 확인하시기 바랍니다.

본 제품을 사용하시다가 생긴 손해 및 손실, 간접적인 손상, 재산상의 손실, 부상또는 사망에 대하여 저희 컴파일 테크놀로지 주식회사는 책임이 없음을 명시하는 바입니다. 본 문서에 사용된 예제, 삽화, 도표, 데이터 등은 설명을 위한 예시적인 자료일 뿐입니다. 이 문서에 제시된 예시 및 데이터의 사용에 따른 책임은 소비자 여러분에게 있음을 명시합니다. 이에 대한 동의가 없다면 본 제품 사용을 중단하여 주시기바랍니다.

차례

제 1 장 제품설명	6
제품 소개	7
하드웨어 설명	9
과워옵션 보드	11
LAYER.....	13
좌표 시스템	14
초기화	16
폰트와 TEXT.....	19
다국어 폰트 지원.....	20
제 2 장 프로토콜	21
UTF-8 프로토콜	22
응답 멧세지.....	23
텍스트 작성 방법.....	24
UTF-8 모드 체크섬 사용시.....	25
BINARY 프로토콜	26
응답 멧세지.....	28
Binary 모드 체크섬 사용시.....	29
제 3 장 그래픽코멘드	31
ARCTO (CODE : 00).....	32
BACKLIGHT (CODE : 01).....	34
BEVELBUTTON (CODE : 02)	35
BUTTON (CODE : 03)	36
CLEAR (CODE : 04).....	37
CLEARPATH (CODE : 05).....	38
CLOSEPATH (CODE : HEX 32).....	39
COLOR (CODE : 07)	40
CREATELAYER (CODE : 08)	42
CURVETO (CODE : 09).....	43
DELAY (CODE : HEX 0A).....	44
DESTORYLAYER (CODE : HEX 0B).....	45

DISABLEFLUSH (CODE : HEX 0C).....	46
DISABLELAYER (CODE : HEX 34).....	47
DRAWARC (CODE : HEX 0D).....	48
DRAWCIRCLE (CODE : HEX 0E)	49
DRAWCURVE (CODE : HEX 0F)	50
DRAWELLIPSE (CODE : HEX 10).....	52
DRAWLINE (CODE : HEX 11).....	53
DRAWPATH (CODE : HEX 12).....	54
DRAWRECTANGLE (CODE : HEX 13).....	55
ENABLEFLUSH (CODE : HEX 33)	57
ENABLELAYER (CODE : HEX 35).....	58
FILLCIRCLE (CODE : HEX 16).....	59
FILLELLIPSE (CODE : HEX 17).....	60
FILLPATH (CODE : HEX 18).....	61
FILLRECTANGLE (CODE : HEX 19)	62
FILLTEXT (CODE : HEX 1A)	63
FONTSIZE (CODE : HEX 1B)	66
GLOSSYBUTTON (CODE : HEX 1C).....	67
IMAGE (CODE : HEX 1D)	68
LAYER (CODE : HEX 1E).....	69
LINECAP (CODE : HEX 1F)	70
LINEJOIN (CODE : HEX 20).....	71
LINETo (CODE : HEX 21).....	72
LINEWIDTH (CODE : HEX 22).....	73
LINEARGRAIDENT (CODE : HEX 23).....	74
LOADFONT (CODE : HEX 24)	75
MOVELAYER (CODE : HEX 25).....	76
MOVETO (CODE : HEX 26)	77
PING (CODE : HEX 27)	78
RADIALGRADIENT (CODE : HEX 28)	79
RUNSCRIPT (CODE : HEX 29)	80
RESET (CODE : HEX 2A).....	83
SETPIXEL (CODE : HEX 2C).....	84
UNLOADFONT (CODE : HEX 2D).....	85
제 4 장 라이브러리.....	87

MOACON 용 라이브러리.....	88
부록	91
폰트 서브셋 유틸리티.....	92
펌웨어 업그레이드	93
M DISPLAY 콘솔 프로그램.....	95
M DISPLAY TEST 프로그램.....	98

제 1 장

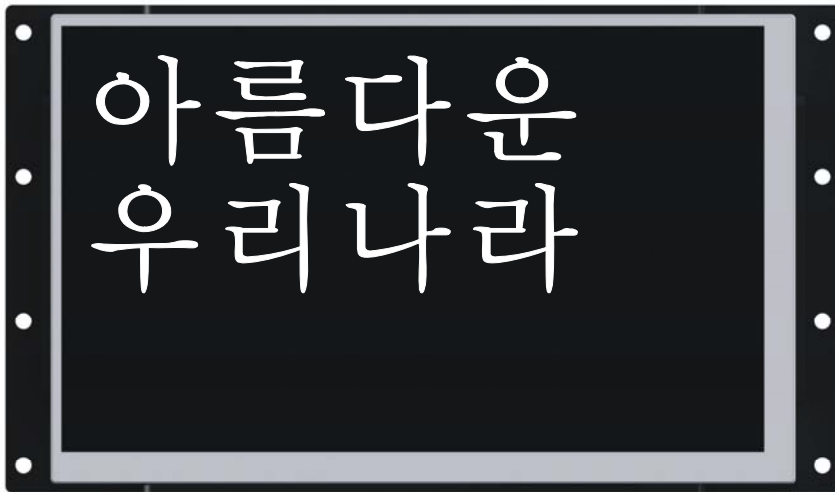
제품설명

제품 소개

“M 디스플레이”는 RS232C 접속만으로 간편하게 LCD 상에 그래픽과 한글을 표시할 수 있는 제품입니다.

RS232 인터페이스가 있는 기기 (PC 또는 MCU, 모아콘 등)만 있다면, 쉽게 구동하실 수 있습니다.

WINDOWS 에서 사용하는 트루타입 폰트를 그대로 사용할 수 있어서, 글자 크기와 관계없이, 미려한 폰트를 사용할 수 있다는 것이 장점입니다.



또한 원하는 배경이미지를 PC 상에서 포토샵등으로 작업한뒤 PNG 파일로 마이크로 SD 카드에 저장하여, LCD 상에 그대로 표시할 수 있습니다.

*큐블록은 M Display 용으로 적합하지 않습니다. 모아콘을 사용하시기 바랍니다. M Display 구동을 위한 모아콘용 라이브러리를 제공하고 있습니다. (제 4 장 라이브러리 참조)

주요 기능

- 고해상도 TFT 칼라 디스플레이
- 트루타입 한글/영문 폰트 지원 (다국어 지원)
- RS232C 인터페이스
- 다양한 그래픽 코멘드 지원 (선, 원, 사각형, 커브, 이미지등)
- 알파블렌딩 지원 (이미지를 투명하게 만들어주는 기능)
- PNG 이미지 파일 표시가능
- 마이크로 SD 카드 지원 (폰트, 이미지, 스크립트 파일 기록)
- USB 로 펌웨어 업그레이드 가능
- 터치입력 지원안함

주요 사양

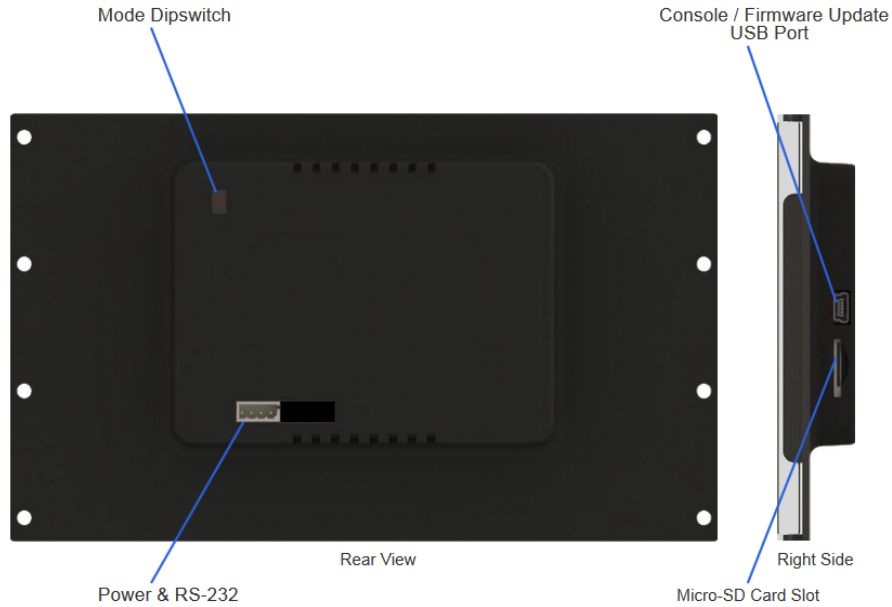
- 16 비트 칼라
- 4 메가 램
- 168MHz ARM 프로세서 채용
- RS232C 포트 1 채널 내장
- 마이크로 SD 카드 슬롯, USB 포트내장

제품 구성품 : 파워 옵션보드 과 4 핀 케이블, MDisplay 모듈 , 2G SD 카드 (SD 카드에는 한글폰트파일이 저장되어 있습니다.)

모델

모델명	화면크기	해상도
MDP070N	7 인치	800 X 480
MDP056N	5.6 인치	640 X 480
MDP043N	4.3 인치	480 X 272
MDP102N	10.2 인치	800 X 480

하드웨어 설명



모드 스위치

딤스위치 1 (데모모드)	ON : 데모화면 표시 무한반복 OFF : 데모 모드 표시안함
딤스위치 2 (펌웨어 업그레이드)	ON : 펌웨어 업그레이드 모드 진입 자세한 내용은 “펌웨어 업그레이드방법”을 참조하십시오. OFF : RUN 모드 , 정상시는 OFF 로 두십시오.

딤스위가 둘 다 OFF 위치에 있을 때, RS232C 로 부터오는 커멘드를 처리합니다. RS232C 로부터 커멘드를 받지 못하면, 화면상에는 아무것도 표시되지 않습니다.

4 핀 콘넥터

PIN1	RS232C TX	M 디스플레이가 송신합니다.
PIN2	RS232C RX	M 디스플레이가 수신합니다.
PIN3	POWER (+5V)	5V 전원 공급단자
PIN4	GND	그라운드

5V 전원은 반드시 1A 이상 공급이 가능한 파워 서플라이를 사용하십시오. 본 제품은 대략 800mA의 전류를 소모합니다.

마이크로 SD 슬롯

폰트파일, PNG 이미지파일, 스크립트 파일등은 MICRO SD 카드에 저장합니다.

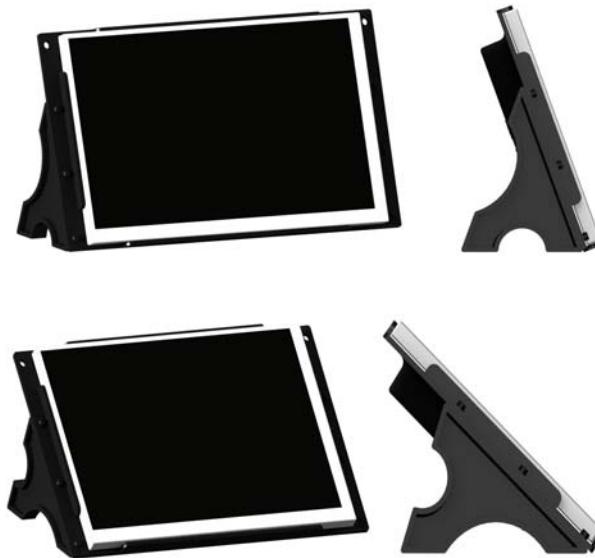
- FAT32 포맷만 지원합니다.
- 긴 파일명을 지원하지 않습니다. 최대 8자 이내로 파일명을 작성하세요.

USB 포트

펌웨어 업그레이드 또는 모니터링 정보 표시를 위한 USB 포트입니다.

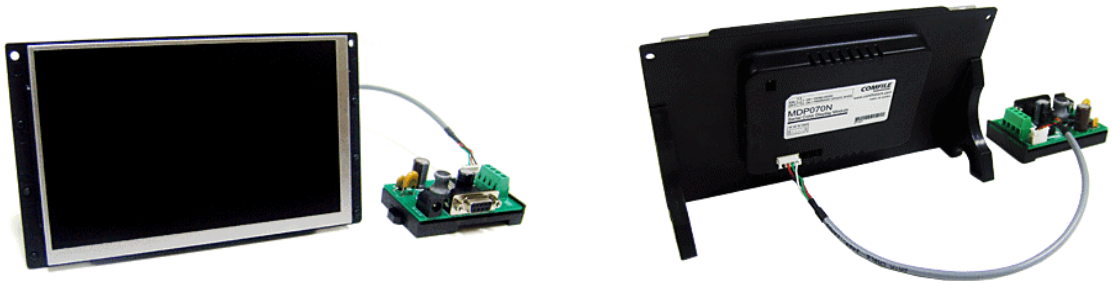
스탠드

개발시 편리하도록 책상위에 세울수 있는 스탠드를 제공합니다. 2가지 각도중 하나를 선택하실 수 있습니다.



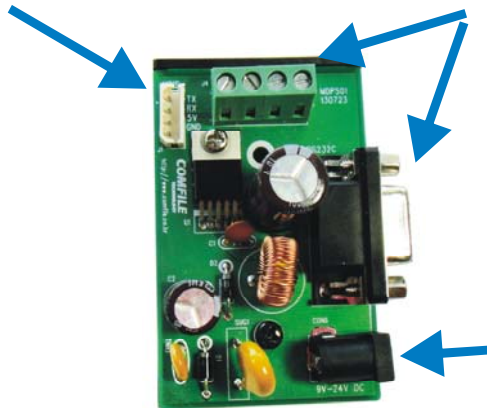
파워옵션 보드

M DISPLAY 는 5V 전원으로 구동되도록 되어 있습니다. 파워옵션 보드는 9V~24VDC 전원을 인가하면 5V (1A)을 출력해주는 보드입니다. 또한 D-SUB 콘넥터 타입의 RS232C 포트가 있어 PC 와 직접 연결하실 수 있습니다.



M Display 와 연결 (4 핀 케이블 제공됨)

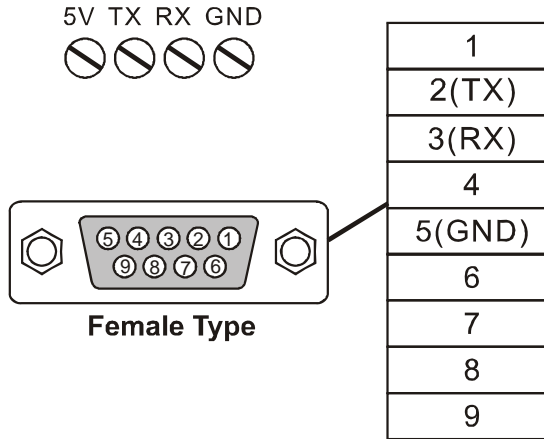
RS232C 입력 (둘중 하나만 사용)



DC9V~24V 전원입력

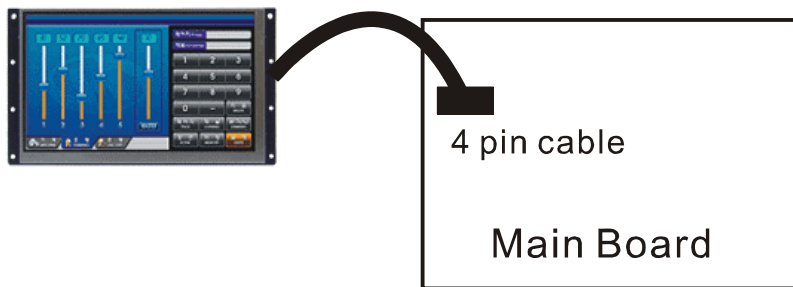
RS232C (+/- 10V 신호)는 DSUB 단자와 터미널 블록중 하나에만 입력하십시오.

PC 와 연결할때에는 DSUB 단자에 1:1 케이블을 사용하십시오.



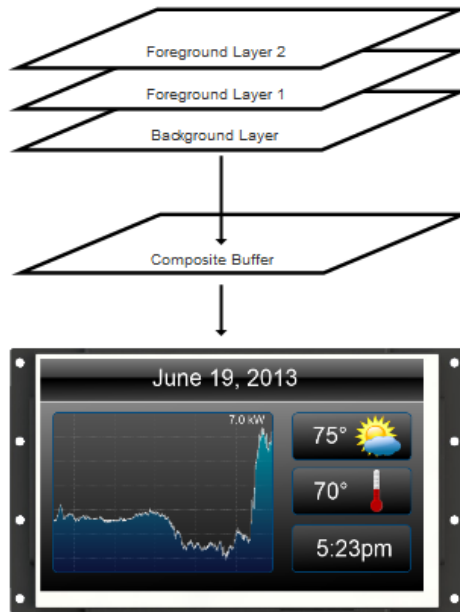
터미널블록의 5V 단자는 내부에서 만들어진 5V 가 출력되는 단자입니다.

파워 옵션 보드는 개발시 편리하도록 제공되는 보드입니다. 실제 제품에서는 여러분이 설계한 메인보드와 직접 연결되도록 하십시오. (5V 는 반드시 1A 이상이 공급되어야 합니다.)



Layer

MDisplay에서는 Layer를 지원합니다. 여러장의 Layer가 존재하고, 이 Layer가 모두 합쳐져서 보여지는 것을 말합니다.

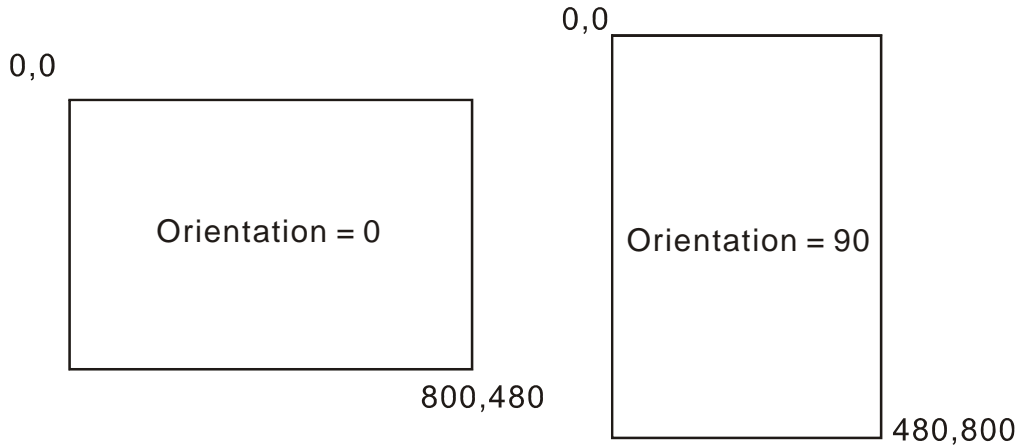


과워온시 Background Layer가 자동 생성되고, 이것이 디폴트 Layer가 됩니다. 이후 “CreateLayer” 명령에 의해 Layer를 추가하실 수 있습니다.

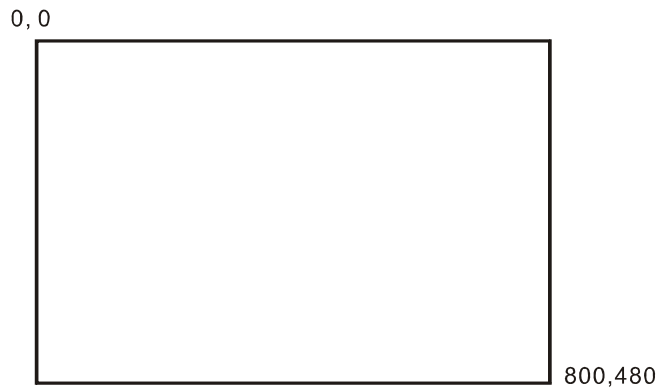
Layer 명령에 의해서 이중 하나를 Active할 수 있으며, 모든 그래픽 코멘드는 Active된 Layer에 결과를 표시합니다.

좌표 시스템

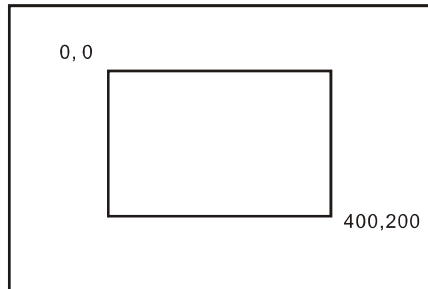
MDP070N 모델의 경우 가로방향에서는 800 x 480, 세로방향에서는 480 x 800 해상도를 지원합니다.



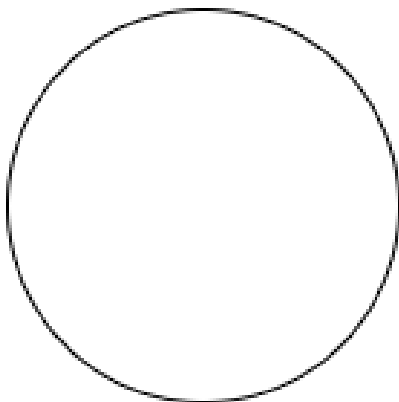
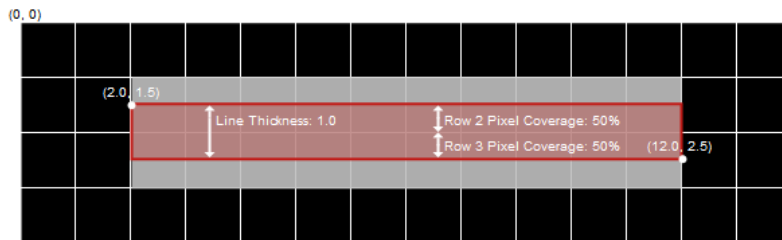
좌표 기준점은 Active 된 Layer 의 영향을 받게 됩니다. 최초 Background Layer 만 Active 된 상태에서는 화면가장 왼쪽 위가 0,0 지점이 됩니다.



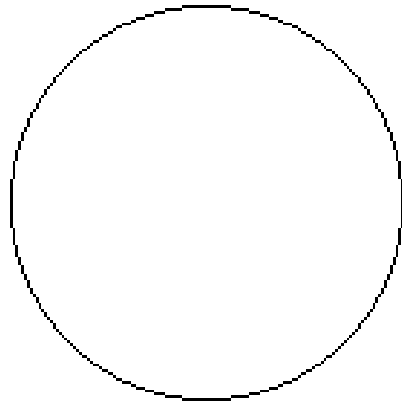
400, 200 사이즈의 Layer 를 화면 중앙에 위치시키고, Active 하면 아래와 같이 Active 된 Layer 의 좌측상단이 0,0 이 됩니다.



일부 코멘드에서 좌표는 소수점 이하 1 자리까지 사용되는데, 그 이유는 좀 더 부드러운 곡선처리를 위한 “안티 알리아싱”을 구현하기 위해서 입니다.



Anti-aliased



Aliased

초기화

MDisplay 는 파워온시 디폴트로 다음과 같은 상태가 됩니다.

종류	파워온시 상태
통신 모드	115200 보레이트, 8 비트, No 패리티, 1 스톱비트
프로토콜	UTF-8
체크섬	사용안함
표시방향	가로
부팅이미지	없음

이것을 다른 상태로 바꾸기 위해서는, SD 카드의 루트(root)에 “init.txt”라는 TEXT 파일을 작성하고, 그안에 다음과 같은 코멘트를 적어주어야 합니다.

PC 상에서 <메모장>과 같은 TEXT EDITOR 프로그램을 이용해서 init.txt 파일을 만들고, 그것을 SD 카드에 카피한뒤, M DISPLAY 의 SD CARD 슬롯에 넣어주십시오.

Serial

```
Serial baudRate parity stopBits
```

통신모드를 결정하는 명령어 입니다.

Argument	Description
baudRate	1200, 2400, 4800, 9600, 57600, 115200 중 하나
parity	“odd” “even”, “none” 중 하나
stopBits	1.0, 2.0 중 하나

Protocol

```
Protocol protocol
```

프로토콜을 결정하는 명령어 입니다.

Argument	Description
protocol	“UTF-8” “Binary” 중 하나

EnableChecksum

```
EnableChecksum
```

이 명령어가 init.txt 안에 있다면, 체크섬 사용모드로 전환됩니다.

EnableTrace

```
EnableTrace
```

이 명령어가 init.txt 안에 있다면, 명령어 수신상태가 콘솔프로그램쪽으로 출력됩니다. 자세한 설명은 <부록>에 있는 <콘솔 프로그램>설명을 참조하시기 바랍니다.

Orientation

```
Orientation orientation
```

표시방향을 결정합니다. 90 도나 270 도로 되어 있으면, 세로방향으로 표시를 합니다.

Argument	Description
protocol	0, 90, 180, 270 중 하나

BootImage

```
BootImage filePath
```

부팅시 디폴트로 보여줄 png 파일경로와 파일명을 적어줍니다. 전원 ON 시 해당파일이 바로 표시됩니다.

Argument	Description
filePath	이미지파일명 (png 파일)

다음은 init.txt 작성 예입니다.

```
Serial 57600 none 1.0
Protocol UTF-8
EnableChecksum
Orientation 90
BootImage "comfilelogo.png"
```

이 init.txt 가 sd 카드의 root 폴더에 있으면, M DISPLAY 는 다음과 같이 동작합니다.

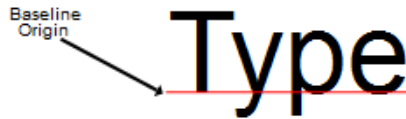
1. 보레이트는 57600, 패리티는 none, 스톱비트는 1
2. 프로토콜은 UTF-8
3. 체크섬은 사용, Trace 는 사용안함
4. 방향은 90 도 회전
5. 파워온시 "comfilelogo.png"를 화면에 표시

폰트와 Text

M Display 는 트루타입 폰트를 지원합니다. LoadFont 명령으로 SD 카드에 있는 폰트파일 (ttf 확장자)중 하나를 선택합니다.

M Display 에는 기본적으로 영문은 Sans-Serif 폰트가 내장되어 있습니다만, 한글폰트는 내장되어 있지 않습니다. LoadFont 명령어로 SD 카드에 있는 한글 폰트를 선택한뒤 한글을 사용할 수 있습니다.

Text 를 표시할 때에는 Baseline 좌표를 사용하여 위치를 정합니다.



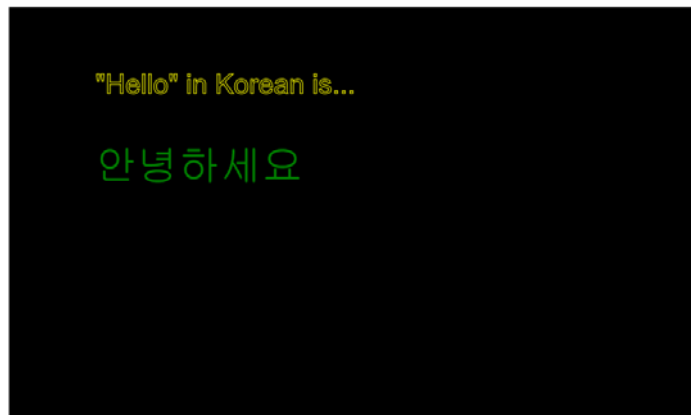
다음은 Text 표시예입니다.

```
# Clear the active layer
Clear

# Draw yellow text in the default font
Color FFFF00
FillText 100 100 "\"Hello\" in Korean is..."

# Load font with Korean glyphs
LoadFont "/gulim.ttf"

# Draw green text in the gulim font
Color 00FF00
FontSize 48
FillText 100 200 "안녕하세요"
```



다국어 폰트 지원

M Display 는<UTF-8 유니코드>를 지원합니다. 따라서 해당국가에서 사용하는 폰트파일을 사용하면, 다른나라의 언어도 화면에 표시할 수 있습니다.

*주의사항: EUC-KR 코드 (일명: 완성형 한글코드)는 사용할 수 없습니다.

다음은 일본어와 중국어를 표시한 예입니다.

```
# English text using default font
FontSize 48
FillText 100 200 "Hello"

# Load font with Korean glyphs
LoadFont "/korean.ttf"
FillText 100 200 "안녕하세요"

# Load font with Japanese glyphs
LoadFont "/japan.ttf"
FillText 100 200 "おはよう"

# Load font with Chinese glyphs
LoadFont "/china.ttf"
FillText 100 200 "你好"
```

Hello

안녕하세요

おはよう

你好

제 2 장

프로토콜

M Display 는 두 종류의 프로토콜중 하나를 사용할 수 있습니다.

UTF-8 모드	ASCII 호환모드, 모든 코멘드와 좌표가 ASCII 코드로 되어 있습니다.
Binary 모드	이진 모드, 모든 코멘드와 좌표가 2 진숫자로 되어 있습니다.

이중 어떤 프로토콜을 사용할지는 init.txt 에서 결정할 수 있습니다. <초기화>설명참조

UTF-8 프로토콜

UTF-8 프로토콜의 모든 명령어는 다음과 같은 포맷을 갖추고 있습니다.

```
<\n><본체><\r>
```

\n = 16 진수 0A

\r = 16 진수 0D

사용예)

```
\nDrawLine 0 0 100 100\r
```

명령어는 대소문자 구분을 하지 않습니다. 인수(Arguments)는 다음과 같은 type 이 사용됩니다.

Argument Type	Format/Notation	Description
Real Number	실수 (3.4E +/- 38)	32-bit 실수 예) 10.3 10 0 320.5
UInt8	10 진수 (0 ~ 255)	8-bit 부호없는 정수
Int16	10 진수 (-32,768 ~ 32,767)	16-bit 부호있는 정수
UInt16	10 진수 (0 ~ 65,535)	16-bit 부호없는 정수
UInt32	10 진수 (0 ~ 4,294,967,295)	32-bit 부호없는 정수
Color	16 진수 (AARRGGBB). 예: 불투명 빨강 = 00FF0000	32 비트 부호없는 정수 Alpha(AA, 00 = 불투명, FF = 투명), Red(RR), Green(GG), Blue(BB)
Text	UTF-8	UTF-8 코드로된 문자열

COLOR 는 24 비트로 표현하지만, 내부적으로는 16 비트 (5,6,5 비트)로 처리됩니다.

응답 메시지

M DISPLAY 는 모든 명령어에 대해 Status code 를 포함한 응답 메시지를 보내줍니다.

Status Code (Hexadecimal)	Description
0000	Success
0001	Unknown command
0002	Wrong number of command arguments
0003	Invalid command argument
0004	Error reading from a file
0005	Memory allocation error
0006	Serial communication error
0007	Checksum error
FFFF	Unspecified error

다음과 같은 포맷을 응답 메시지를 보내줍니다.

성공시 Ack 라는 문자를 포함하고 있습니다.

```
\nAck 0000\r
```

비정상적인 실행인 경우에는 Nak 라는 문자를 포함하고 있습니다.

알 수 없는 코멘드 일 때 다음과 같이 응답합니다.

```
\nNak 0001\r
```

텍스트 작성 방법

UTF-8 프로토콜에서 텍스트 작성 방법은 다음과 같습니다.

다음 명령을 실행하면 결과가 다음과 같이 표시됩니다.

```
FillText 100 100 Hello World
```

Hello

문자열이 공백을 포함하고 있다면 쌍따옴표로 감싸주어야 합니다.

```
FillText 100 100 "Hello World"
```

Hello World

문자열중에 쌍따옴표를 표시하고 싶다면 다음과 같이 하십시오.

```
FillText 100 100 "Double Quotes (\") must be escaped"
```

Double Quotes (") must be escaped

문자열중에 역슬래시를 표시하고 싶다면 다음과 같이 하십시오.

```
FillText 100 100 "Backslashes (\\) must also be escaped"
```

Backslashes (\) must also be escaped

UTF-8 모드 체크섬 사용시

체크섬 On 모드에서는 다음과 같은 방법으로 모든 명령문 끝에 체크섬을 추가해서 보내주어야 합니다.

명령문안에 모든 문자의 UTF-8 (ASCII)코드값을 서로 더한뒤 맨뒤에 그 값을 적어줍니다.

(D + r + a + w + L + I + n + e + space + 0 + space + 0 + space + 1 + 0 + 0 + space + 1 + 0 + 0 + space) 모두 더한결과 = 16 진수 0538

```
DrawLine 0 0 100 100 0538
```

응답 메시지도 체크섬을 포함하고 있습니다.

(A + c + k + space + 0 + 0 + 0 + 0 + space) 모두 더한결과 = 16 진수 020F

```
Ack 0000 020F
```

Binary 프로토콜

Binary 프로토콜의 모든 명령어는 다음과 같은 포맷을 갖추고 있습니다.

<DLE><STX><본체><DLE><ETX>

<DLE> = 16 진수 10
 <STX> = 16 진수 02
 <ETX> = 16 진수 03

사용예)

항목	DLE	STX	DrawLine	0	0	1000	1000	DLE	ETX
Byte(Hex)	10	02	11	00-00	00-00	E8-03	E8-03	10	03

16 비트, 또는 32 비트 수를 보낼때에는 아랫쪽 바이트를 먼저 보내주어야 합니다. 즉, 3E8 을 보내려면 E8 을 먼저보내고 03 을 나중에 보내야 합니다.

인수(Arguments)는 다음과 같은 type 이 사용됩니다.

Argument Type	Description
Real Number	10 을 곱한 좌표. 예: 123.4 는 1234 로 써줍니다.
UInt8	8-bit 부호없는 정수
Int16	16-bit 부호있는 정수
UInt16	16-bit 부호없는 정수
UInt32	32-bit 부호없는 정수
Color	부호가 없는 32-bit 정수 , 16 진수 (AARRGGBB) Alpha(AA, 00 = 불투명, FF = 투명), Red(RR), Green(GG), Blue(BB) 예) 000000FF = 불투명 파랑 0000FF00 = 불투명 녹색 00FF0000 = 불투명 빨강
Text	문자열, 맨뒤에는 NULL 문자가 필요합니다.

이중 Real Number 는 소수점 밑 한자리를 포함하는 좌표를 뜻합니다. 123.4 를 원하면 위해서는 1234 를 사용합니다. 즉 실제값에서 곱하기 10 한 값을 사용하는 것입니다.

COLOR 는 24 비트로 표현하지만, 내부적으로는 16 비트 (565 비트)로 처리됩니다.

DrawLine 에 해당하는 코멘드 코드인 11 을 사용하였습니다.

항목	DLE	STX	코멘드 code (DrawLine)	0	0	1000	1000	DLE	ETX
Byte(Hex)	10	02	11	00-00	00-00	E8-03	E8-03	10	03

본체에 <DLE>코드값인 10 이 포함되어 있다면, <DLE><DLE>로 보내주어야 합니다. 즉 10 을 두번 연속 보내주어야 합니다.

항목	DLE	STX	DrawLine	16	0	1000	1000	DLE	ETX
Byte(Hex)	10	02	11	10-10-00	00-00	E8-03	E8-03	10	03

문자열 뒤에는 반드시 00 을 보내야 합니다.

항목	DLE	STX	FillText	1000	1000	Hello	DLE	ETX
Byte(Hex)	10	02	1A	E8-03	E8-03	48-65-6C-6C-6F- 00	10	03

응답 메시지

M DISPLAY 는 모든 명령어에 대해 Status code 를 포함한 응답 메시지를 보내줍니다.

Status Code (Hexadecimal)	Description
0000	Success
0001	Unknown command
0002	Wrong number of command arguments
0003	Invalid command argument
0004	Error reading from a file
0005	Memory allocation error
0006	Serial communication error
0007	Checksum error
FFFF	Unspecified error

다음과 같은 포맷을 응답 메시지를 보내줍니다.

성공시 ACK (HEX 06)을 포함하고 있습니다.

항목	DLE	STX	ACK	성공	DLE	STX
Byte(Hex)	10	02	06	00-00	10	03

비정상적인 실행의 경우 NAK(HEX 15)를 포함하고 있습니다.

항목	DLE	STX	NAK	알수없는 명령어	DLE	STX
Byte(Hex)	10	02	15	01-00	10	03

Binary 모드 체크섬 사용시

체크섬 On 모드에서는 다음과 같은 방법으로 체크섬을 추가해서 보내주어야 합니다.

<DLE><STX>를 제외한 모든 Byte 를 합산하여 사용합니다.

(11 + 00 + 00 + 00 + 00 + E8 + 03 + E8 + 03 = 01E7)

항목	DLE	STX	DrawLine	0	0	1000	1000	Checksum	DLE	STX
Byte(Hex)	10	02	11	00-00	00-00	E8-03	E8-03	E7-01	10	03

응답 메시지도 체크섬을 포함하고 있습니다.

(06 + 00 + 00 + 00 + 00 = 0006)

항목	DLE	STX	ACK	성공	Checksum	DLE	STX
Byte(Hex)	10	02	06	00-00-00-00	06-00	10	03

제 3 장

그래픽코멘드

1. 모든 그래픽 관련 명령은 현재 Active 되어 있는 Layer 에 표시됩니다.
2. 칼라는 Color 명령에 의해 사전에 지정된 칼라가 적용됩니다.
3. 그라데이션 상태는 LinearGradient 또는 RadialGradient 명령어로 사전 지정된 상태를 적용받습니다.
4. 기타 선상태는 LineWidth, LineCap, LineJoin 명령어로 사전 지정된 상태를 적용받게 됩니다.

(Optional)이라고 표시된 Argument 는 생략가능한 Argument 입니다.

ArcTo (code : 00)

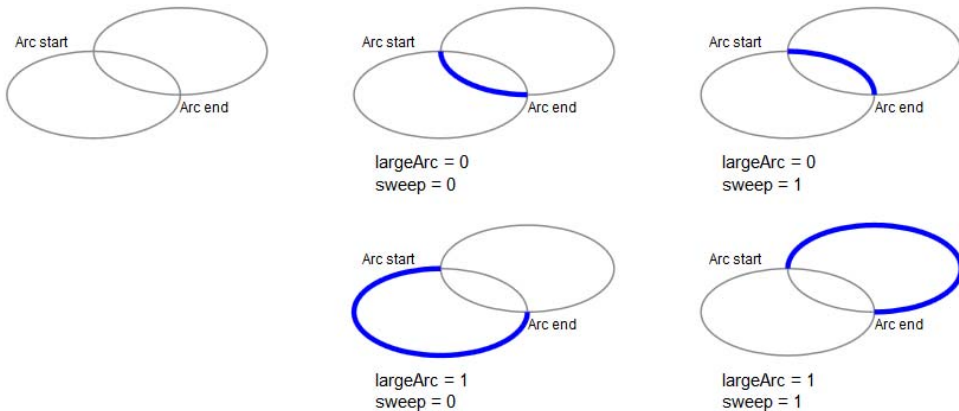
```
ArcTo x y [horizontalRadius [verticalRadius [rotationAngle [largeArc [sweep]]]]]
```

현재 Point 로부터 원 또는 타원의 호(Path)를 생성합니다. 최초 Point 는 Moveto 명령어로 지정할 수 있습니다.

Argument	Type	Description
x	Real Number	X 좌표
Y	Real Number	Y 좌표
horizontalRadius	Real Number	수평반지름
verticalRadius	Real Number	(Optional) 수직 반지름, 생략시 기본값은 수평값과 동일한 값입니다.
rotationAngle	Real Number	(Optional) 회전각, 생략시 기본값은 0 도입니다.
LargeArc	UInt8	(Optional) 0 = 작은쪽 호, 0 이외 값 = 큰 호, 생략시 기본값은 큰쪽 호.
sweep	UInt8	(Optional) 0 = 시계반대방향, 0 이외 값 = 시계방향, 생략시 기본값은 시계방향.

Path 란 path 관련 그리기 명령어 (예: ArcTo, ClearPath, ClosePath, CurveTo, LineTo, MoveTo) 에서 사용하는 <그림을 그려야할 경로>를 뜻합니다. Path 는 바로 화면에 표시되지는 않습니다. 추후 Drawpath 명령을 실행하면 모든 Path 가 화면에 표시됩니다.

LargeArc 와 sweep 설정에 따라 표시되는 호의 위치와 방향을 선택할 수 있습니다.



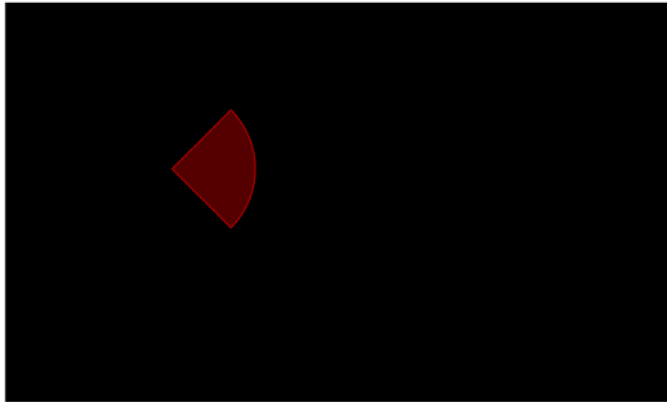
다음은 파이형 차트를 ArcTo 명령을 사용해서 그린 예제입니다.

```
# Clear the active layer
Clear

# Create a fraction of a pie
MoveTo 200 200
LineTo 270.7 129.29
ArcTo 270.7 270.7 100 100 0 0 1
ClosePath

# Fill the pie with partially transparent red
Color AAFF0000
FillPath

# Draw a border around the pie in bright red
LineWidth 1
Color FF0000
DrawPath
```



Backlight (code : 01)

Backlight brightness

Argument	Type	Description
<i>brightness</i>	UInt8	0 부터 255 사이의 값, 0 은 OFF

백 라이트의 밝기를 조절하는 명령입니다. 디폴트값은 255 입니다.

BevelButton (code : 02)

```
BevelButton topLeftX topLeftY width height buttonColor topLeftColor bottomRightColor text
```

입체 느낌이 나는 버튼을 표시합니다.

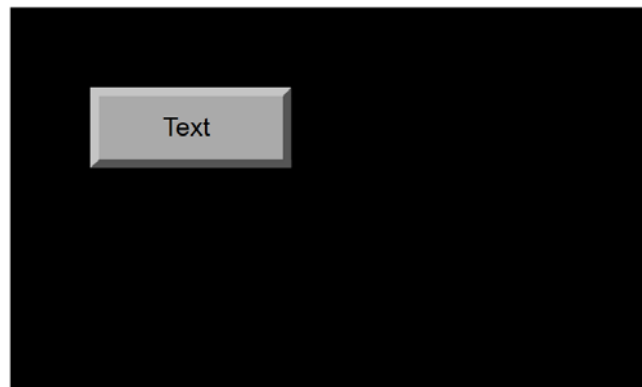
Argument	Type	Description
<i>topLeftX</i>	Real Number	시작 X 좌표
<i>topLeftY</i>	Real Number	시작 Y 좌표
<i>width</i>	Real Number	폭
<i>height</i>	Real Number	높이
<i>buttonColor</i>	Color	버튼의 색
<i>topLeftColor</i>	Color	위쪽과 왼쪽의 색
<i>bottomRightColor</i>	Color	아래쪽과 오른쪽의 색
<i>text</i>	Text	(Optional) 버튼의 정중앙에 표시할 문자

```
# Clear the active layer
Clear

# Black Text
Color 0

# Border width of 10 pixels
LineWidth 10

# Gray beveled button
BevelButton 100 100 250 100 AAAAAA C5C5C5 555555 "Text"
```



Button (code : 03)

```
Button topLeftX topLeftY width height cornerRadius buttonColor borderColor [text]
```

사각형 (또는 모서리가 둥근 사각형) 버튼을 표시합니다.

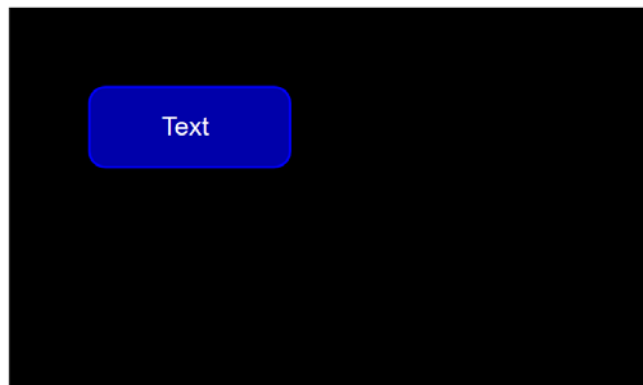
Argument	Type	Description
<i>topLeftX</i>	Real Number	시작 X 좌표
<i>topLeftY</i>	Real Number	시작 Y 좌표
<i>width</i>	Real Number	폭
<i>height</i>	Real Number	높이
<i>cornerRadius</i>	Real Number	버튼 코너 둥근모서리의 반지름
<i>buttonColor</i>	Color	버튼 색
<i>borderColor</i>	Color	테두리 색
<i>text</i>	Text	(Optional) 버튼 정중앙에 표시할 문자

```
# Clear the active layer
Clear

# white text
Color FFFFFFFF

# Border width of 1 pixel
LineWidth 3

# Blue button with 20 pixel radius rounded corners
Button 100 100 250 100 20 0000AA 0000FF "Text"
```



Clear (code : 04)

```
Clear [x y width height]
```

화면을 지우는 코멘드입니다. 파라미터없이 쓴다면, 현재 Active 된 layer 를 전부 클리어합니다.
X, y, width, height 를 적어주면, 현재 Active 된 Layer 의 해당 사각형 안을 클리어합니다.

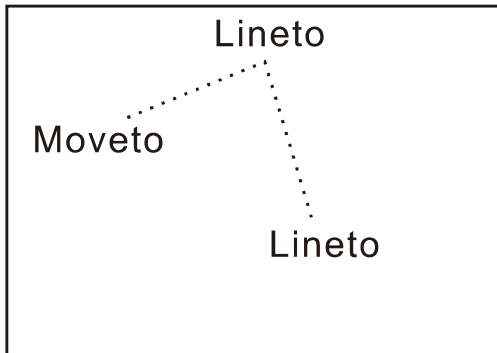
Argument	Type	Description
<i>x</i>	Int16	시작 X 좌표
<i>y</i>	Int16	시작 Y 좌표
<i>width</i>	UInt16	폭
<i>height</i>	UInt16	높이

ClearPath (code : 05)

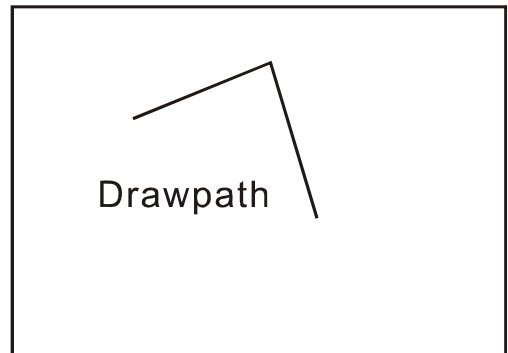
ClearPath

현재 path 를 모두 지웁니다. Path 란 path 관련 그리기 명령어 (예: ArcTo, ClearPath, ClosePath, CurveTo, LineTo, MoveTo) 에서 사용하는 <그림을 그려야할 경로>를 뜻합니다.

path 관련 그리기 명령어 (예: ArcTo, ClearPath, ClosePath, CurveTo, LineTo, MoveTo) 은 실제 화면에 그림을 바로 그리지 않고, Path 만 생성됩니다. 이후 Drawpath 또는 Fillpath 명령어에 의해서 실제 화면상에 표시됩니다.



점선은 실제로는 보이지 않습니다.



Drawpath 명령실행시 지금까지 만든 Path 가 표시됩니다.

ClosePath (code : hex 32)

ClosePath

현재 (Current) Point 와 시작(Origin) Point 를 연결하는 선 (Path)를 생성합니다.

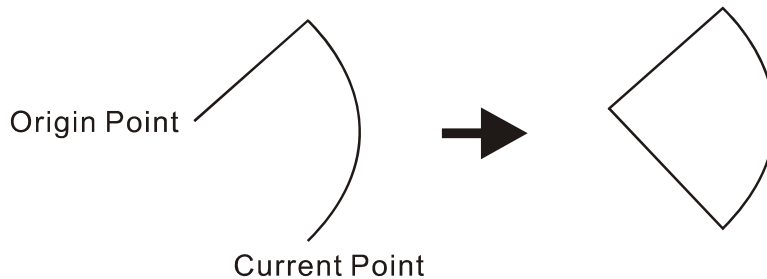
```
# Clear the active layer
Clear

# Create a fraction of a pie
MoveTo 200 200
LineTo 270.7 129.29
ArcTo 270.7 270.7 100 100 0 0 1

# Enclose the pie by drawing a line back to the center
ClosePath

# Fill the pie with partially transparent red
Color AAFF0000
FillPath

# Draw a border around the pie in bright red
LineWidth 1
Color FF0000
DrawPath
```



Color (code : 07)

```
Color color
```

칼라를 변경합니다.

Argument	Type	Description
<i>color</i>	Color	부호가 없는 32-bit 정수 , 16 진수 (AARRGGBB) Alpha(AA, 00 = 불투명, FF = 투명), Red(RR), Green(GG), Blue(BB) 000000FF = 불투명 파랑 0000FF00 = 불투명 녹색 00FF0000 = 불투명 빨강

```
# Clear the active layer
Clear

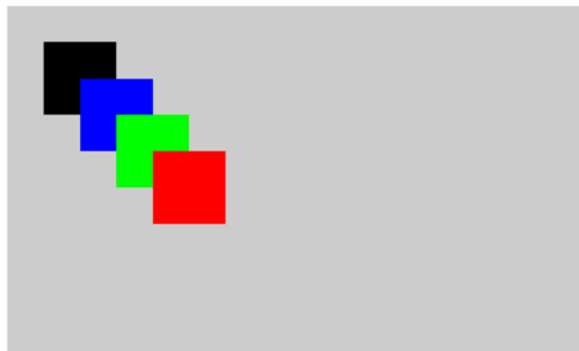
# Opaque gray background
Color CCCCCC
FillRectangle 0 0 800 480

# Opaque black rectangle
Color 0
FillRectangle 50 50 100 100

# Opaque blue rectangle
Color FF
FillRectangle 100 100 100 100

# Opaque green rectangle
Color FF00
FillRectangle 150 150 100 100

# Opaque red rectangle
Color FF0000
FillRectangle 200 200 100 100
```



다음은 부분 투명색으로 겹쳐진 2 개의 사각형을 그리는 것입니다.

```
# Clear the active layer
Clear

# Partially transparent green rectangle
Color A900FF00
FillRectangle 50 50 100 100

# Partially transparent fuchsia rectangle
Color A9FF00FF
FillRectangle 100 100 100 100
```



CreateLayer (code : 08)

```
CreateLayer x y width height layerID
```

새로운 Layer 를 현재 있는 모든 Layer 의 가장 위쪽에 생성합니다.

Argument	Type	Description
<i>X</i>	Int16	시작 X 좌표
<i>Y</i>	Int16	시작 Y 좌표
<i>width</i>	UInt16	폭
<i>height</i>	UInt16	높이
<i>layerID</i>	UInt8	0 을 제외한 숫자, (0=Background Layer)

Layer 는 메모리가 허용하는 한도내에서 추가 생성할 수 있습니다.

Layer 의 사이즈가 작을 경우에는 최대 15 개까지 생성할 수 있습니다. 전체화면 사이즈일 경우에는 1 개만 추가할 수 있습니다. 이 경우 background Layer 까지 포함해서 총 2 개의 Layer 를 사용하실 수 있습니다.

DisableLayer 명령을 사용해서 특정 Layer 를 표시하지 않도록 할 수 있습니다.

EnableLayer 명령을 사용해서 다시 표시하도록 할 수 있습니다.

CurveTo (code : 09)

```
CurveTo x y control1X control1Y [control2X control2Y]
```

현재 포인트로부터 지정된 지점까지 **Bezier** 커브 (Path)를 생성합니다.

Argument	Type	Description
<i>x</i>	Real Number	시작 X 좌표
<i>y</i>	Real Number	시작 Y 좌표
<i>control1X</i>	Real Number	첫번째 콘트롤 포인트의 X 좌표
<i>control1Y</i>	Real Number	첫번째 콘트롤 포인트의 Y 좌표
<i>control2X</i>	Real Number	(Optional) 두번째 콘트롤 포인트의 X 좌표, 두번째 포인트가 있으면 cubic bezier 곡선을 그리고, 없으면 quadratic bezier 곡선을 그립니다.
<i>control2Y</i>	Real Number	(Optional) 두번째 콘트롤 포인트의 Y 좌표, 두번째 포인트가 있으면 cubic bezier 곡선을 그리고, 없으면 quadratic bezier 곡선을 그립니다.

다음은 (200,200) 부터 (400,200)까지 커브를 그린 예입니다.

```
# Clear the active layer
Clear

# Create the vertices for the path
MoveTo 200 200
CurveTo 400 200 300 300 300 200

# Draw the path in a cyan, with a line width of 5 pixels
LineWidth 5
Color 00FFFF
DrawPath
```



Delay (code : hex 0A)

Delay milliseconds

시간을 지연시키는 명령입니다.

Delay 명령은 Script 안에서 그림이 그려지는 동안 대기하는 용도로 사용하게 됩니다.

Script 기능에 대해서는 RunScript 명령을 참조하세요.

Argument	Type	Description
<i>milliseconds</i>	UInt32	잠시 멈출 밀리초 값

DestoryLayer (code : hex 0B)

```
DestroyLayer layerID
```

LayerID 로 지정한 Layer 의 내용을 지웁니다. 해당 Layer 도 더 이상 존재하지 않게됩니다.

Argument	Type	Description
<i>layerID</i>	UInt8	지울 layerID

DisableFlush (code : hex 0C)

DisableFlush

그리는 과정을 숨기기 위한 명령어입니다.

DisableFlush 명령실행뒤 여러가지 그리는 명령어를 실행한다면, 화면상에 표시되지 않고 내부 메모리에만 반영됩니다. 이후 EnableFlush 명령을 실행하면, 최종화면이 바로 표시됩니다.

DisableLayer (code : hex 34)

```
DisableLayer layerID
```

Layer 를 숨깁니다. 다시 표시하려면 EnableLayer 명령을 사용하세요.

Argument	Type	Description
<i>layerID</i>	UInt8	숨길 layerID

DrawArc (code : hex 0D)

```
DrawArc centerX centerY horizontalRadius [verticalRadius] startAngle sweep
```

호를 그립니다.

Argument	Type	Description
<i>centerX</i>	Real Number	타원의 중심 X 좌표
<i>centerY</i>	Real Number	타원의 중심 Y 좌표
<i>horizontalRadius</i>	Real Number	타원의 수평 반지름.
<i>verticalRadius</i>	Real Number	(Optional) 타원의 수직 반지름, 생략되면 수평반지름과 같은값으로 처리됩니다.
<i>startAngle</i>	Real Number	시작 각도
<i>sweep</i>	Real Number	표시할 각도, 음수이면 시계반대방향으로 그립니다.

```
# Clear the active layer
Clear

# Draw green circular arc with a stroke width of 1.0
LineWidth 1.0
Color 00FF00

# Width omitted for circular arc
DrawArc 100 100 50 45.0 235.0

# Draw fuchsia elliptical arc with a stroke width of 3.0
LineWidth 3.0
Color FF00FF
DrawArc 300 100 100 50 110 270
```



DrawCircle (code : hex 0E)

```
DrawCircle centerX centerY radius
```

원을 그립니다.

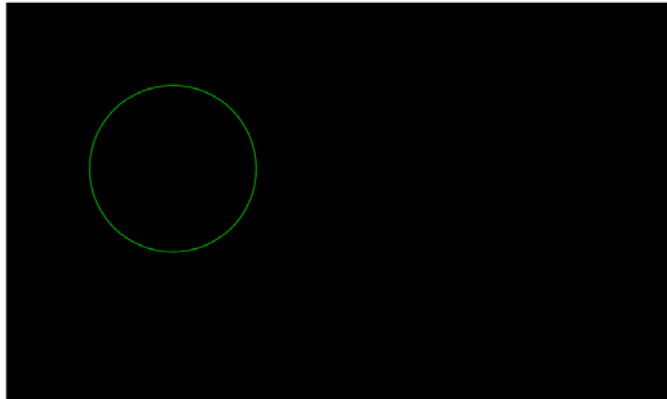
Argument	Type	Description
<i>centerX</i>	Real Number	원의 중심 X 좌표
<i>centerY</i>	Real Number	원의 중심 Y 좌표
<i>radius</i>	Real Number	반지름

다음 예는 좌표 200,200 을 중심으로 하고 반지름이 100 인 원을 그리는 예입니다.

```
# Clear the active layer
Clear

# Set line width
LineWidth 1.0

# Draw a green circle
Color 00FF00
DrawCircle 200 200 100
```



DrawCurve (code : hex 0F)

```
DrawCurve fromX fromY control0X control0Y [control1X control1Y] toX toY
```

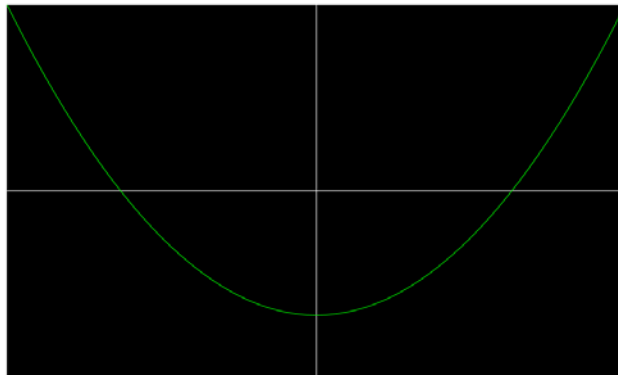
Bezier 곡선을 그립니다.

Argument	Type	Description
<i>fromX</i>	Real Number	시작 X 좌표
<i>fromY</i>	Real Number	시작 Y 좌표
<i>control0X</i>	Real Number	첫번째 콘트롤 포인트의 X 좌표
<i>control0Y</i>	Real Number	첫번째 콘트롤 포인트의 Y 좌표
<i>control1X</i>	Real Number	(Optional) 두번째 콘트롤 포인트의 X 좌표, 두번째 포인트가 있으면 cubic bezier 곡선을 그리고, 없으면 quadratic bezier 곡선을 그립니다.
<i>control1Y</i>	Real Number	(Optional) 두번째 콘트롤 포인트의 Y 좌표, 두번째 포인트가 있으면 cubic bezier 곡선을 그리고, 없으면 quadratic bezier 곡선을 그립니다.
<i>toX</i>	Real Number	종료 X 좌표
<i>toY</i>	Real Number	종료 Y 좌표

```
# Clear the active layer
Clear

# Draw X and Y axes
Color FFFFFFF
DrawLine 0 240 799 240
DrawLine 400 0 400 479

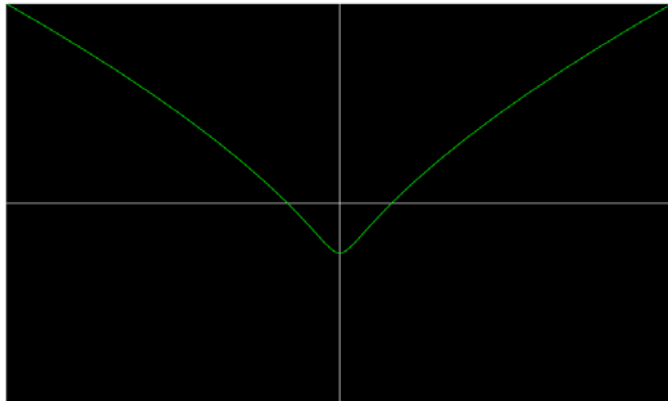
# Draw quadratic bezier curve (parabola)
Color 00FF00
DrawCurve 0 0 400 800 799 0
```



```
# Clear the active layer
Clear

# Draw X and Y axes
Color FFFFFF
DrawLine 0 240 799 240
DrawLine 400 0 400 479

# Draw cubic bezier curve
Color 00FF00
DrawCurve 0 0 700 400 100 400 799 0
```



DrawEllipse (code : hex 10)

```
DrawEllipse centerX centerY horizontalRadius verticalRadius
```

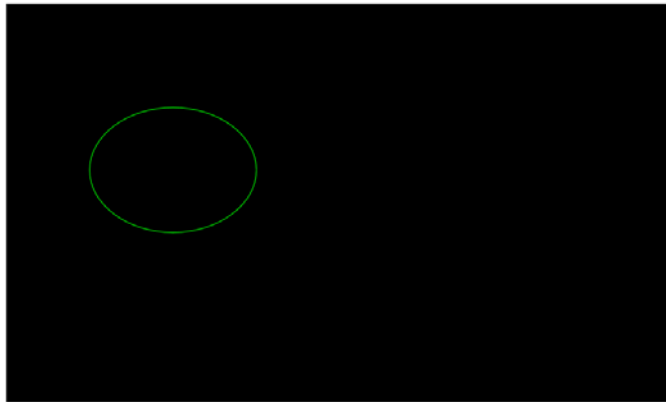
타원을 그립니다.

Argument	Type	Description
<i>centerX</i>	Real Number	타원의 중심 X 좌표
<i>centerY</i>	Real Number	타원의 중심 Y 좌표
<i>horizontalRadius</i>	Real Number	수평 반지름
<i>verticalRadius</i>	Real Number	수직 반지름

```
# Clear the active layer
Clear

# Set line width
LineWidth 1.0

# Draw a green circle
Color 00FF00
DrawEllipse 200 200 100 75
```



DrawLine (code : hex 11)

```
DrawLine x1 y1 x2 y2
```

선을 그립니다.

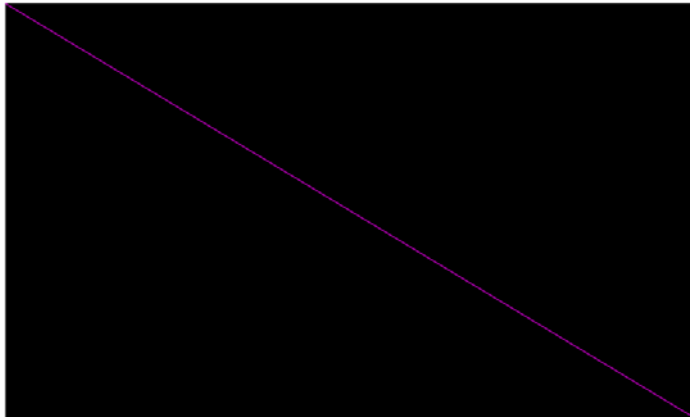
Argument	Type	Description
<i>x1</i>	Real Number	시작 X 좌표
<i>y1</i>	Real Number	시작 Y 좌표
<i>x2</i>	Real Number	끝 지점 X 좌표
<i>y2</i>	Real Number	끝 지점 Y 좌표

```
# Clear the active layer
Clear

# Set the line width
LineWidth 1.0

# Set the color of the line
Color FF00FF

# Draw the line
DrawLine 0 0 799 479
```



DrawPath (code : hex 12)

DrawPath

Path 를 생성하는 모든 그래픽 코멘드 (예를 들면 ArcTo, ClearPath, ClosePath, CurveTo, LineTo, MoveTo)가 만들어놓은 Path 를 화면상에 표시합니다.

```
# Clear the active layer
Clear

# Create the vertices for the path
MoveTo 200 200
CurveTo 400 200 300 300 300 200

# Draw the path in a cyan, with a line width of 5 pixels
LineWidth 5
Color 00FFFF
DrawPath
```



DrawRectangle (code : hex 13)

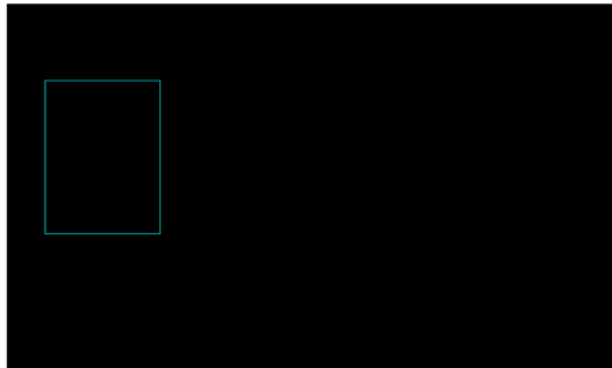
```
DrawRectangle x y width height [topLeftCornerRadius [topRightCornerRadius
bottomLeftCornerRadius bottomRightCornerRadius]]
```

사각형 또는 모서리가 둥근 사각형을 그립니다.

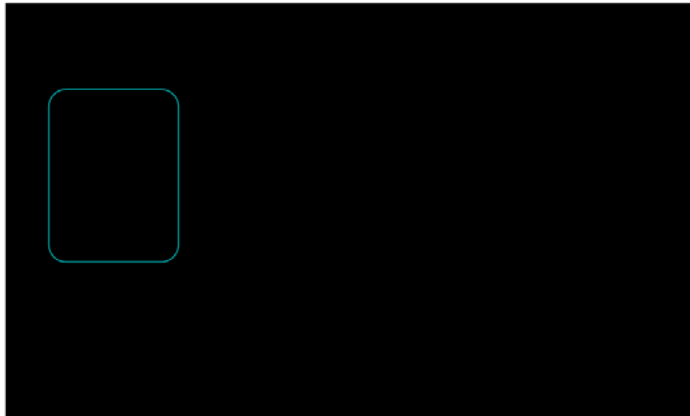
Argument	Type	Description
<i>x</i>	Real Number	시작 X 좌표
<i>y</i>	Real Number	시작 Y 좌표
<i>width</i>	Real Number	폭
<i>height</i>	Real Number	높이
<i>topLeftCornerRadius</i>	Real Number	(Optional) 왼쪽위 모서리 반지름, 생략되면 뾰족한 모서리. 아래 <i>topRightCornerRadius</i> , <i>bottomLeftCornerRadius</i> , <i>bottomRightCornerRadius</i> 이 모두 생략되면 이 값이 다른 코너에도 모두 적용됩니다.
<i>topRightCornerRadius</i>	Real Number	(Optional) 오른쪽 위 모서리 반지름
<i>bottomLeftCornerRadius</i>	Real Number	(Optional) 왼쪽 아래 모서리 반지름
<i>bottomRightCornerRadius</i>	Real Number	(Optional) 오른쪽 아래 모서리 반지름

```
#Clear the current layer
Clear

# Draw a cyan rectangular perimeter
Color 00FFFF
DrawRectangle 50 100 150 200
```



```
#Clear the current layer  
Clear  
  
# Set Line Width  
LineWidth 1.0  
  
# Draw a cyan rounded rectangular perimeter  
Color FFFF  
DrawRectangle 50 100 150 200 20
```



EnableFlush (code : hex 33)

EnableFlush

그리는 과정을 숨기기 위한 명령어 입니다.

DisableFlush 명령실행뒤 여러가지 그리는 명령어를 실행한다면, 화면상에 표시되지 않고 내부 메모리에만 반영됩니다. 이후 EnableFlush 명령을 실행하면, 최종화면이 바로 표시됩니다.

EnableLayer (code : hex 35)

```
EnableLayer layerID
```

Layer 를 표시합니다. CreateLayer 를 한 직후에는 별도로 EnableLayer 명령을 수행하지 않아도 표시됩니다.

Argument	Type	Description
<i>layerID</i>	UInt8	표시할 layerID

FillCircle (code : hex 16)

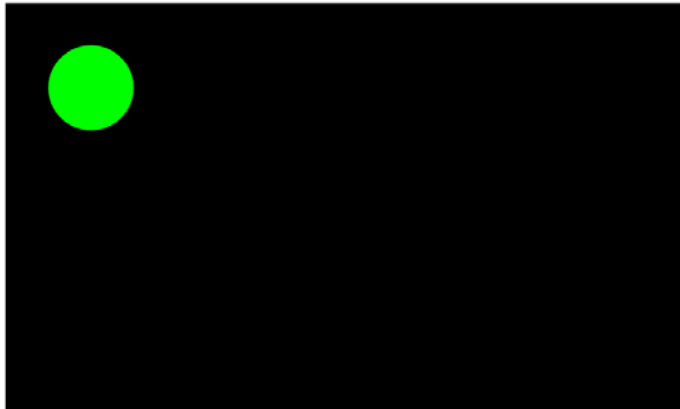
```
FillCircle centerX center, radius
```

채워진 원을 그립니다.

Argument	Type	Description
<i>centerX</i>	Real Number	중심 X 좌표
<i>centerY</i>	Real Number	중심 Y 좌표
<i>radius</i>	Real Number	반지름

```
# Clear the active layer
Clear

# Draw a green circle
Color 00FF00
FillCircle 100 100 50
```



FillEllipse (code : hex 17)

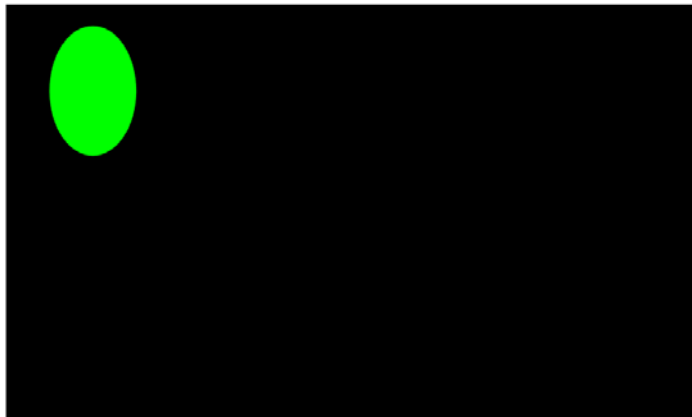
```
FillEllipse centerX centerY horizontalRadius verticalRadius
```

채워진 타원을 그립니다.

Argument	Type	Description
<i>centerX</i>	Real Number	중심 X 좌표
<i>centerY</i>	Real Number	중심 Y 좌표
<i>horizontalRadius</i>	Real Number	수평 반지름
<i>verticalRadius</i>	Real Number	수직 반지름

```
# Clear the active layer
Clear

# Draw a green oval
Color 00FF00
FillEllipse 100 100 50 75
```



FillPath (code : hex 18)

FillPath

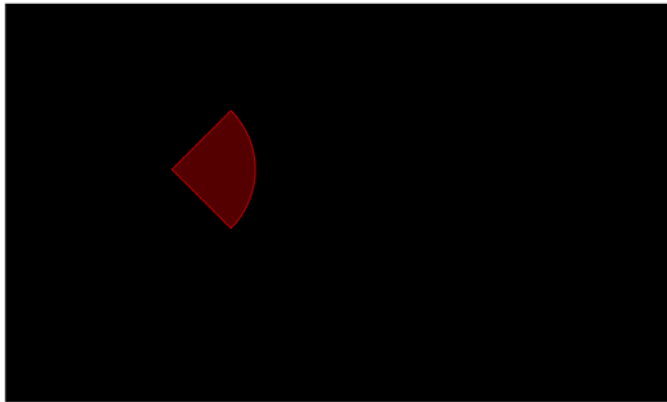
path 관련 명령어 (예: ArcTo, ClearPath, ClosePath, CurveTo, LineTo, MoveTo)로 만들어진 Path 안을 채웁니다. 이때 Path 는 닫힌 폐곡선이여야 합니다.

```
# Clear the active layer
Clear

# Create a fraction of a pie
MoveTo 200 200
LineTo 270.7 129.29
ArcTo 270.7 270.7 100 100 0 0 1
ClosePath

# Fill the pie with partially transparent red
Color AAFF0000
FillPath

# Draw a border around the pie in bright red
LineWidth 1
Color FF0000
DrawPath
```



FillRectangle (code : hex 19)

```
FillRectangle topLeftX topLeftY width height [topLeftCornerRadius [topRightCornerRadius
bottomLeftCornerRadius bottomRightCornerRadius]]
```

채워진 사각형 또는 채워진 모서리가 둥근 사각형을 그립니다.

Argument	Type	Description
<i>topLeftX</i>	Real Number	시작 X 좌표
<i>topLeftY</i>	Real Number	시작 Y 좌표
<i>width</i>	Real Number	폭
<i>height</i>	Real Number	높이
<i>topLeftCornerRadius</i>	Real Number	(Optional) 왼쪽위 모서리 반지름, 생략되면 뾰족한 모서리. 아래 <i>topRightCornerRadius</i> , <i>bottomLeftCornerRadius</i> , <i>bottomRightCornerRadius</i> 이 모두 생략되면 이 값이 다른 코너에도 모두 적용됩니다.
<i>topRightCornerRadius</i>	Real Number	(Optional) 오른쪽 위 모서리 반지름
<i>bottomLeftCornerRadius</i>	Real Number	(Optional) 왼쪽 아래 모서리 반지름
<i>bottomRightCornerRadius</i>	Real Number	(Optional) 오른쪽 아래 모서리 반지름

```
# Clear the active layer
Clear

# Partially transparent green rectangle
Color A900FF00
FillRectangle 50 50 100 100

# Partially transparent fuchsia rectangle
Color A9FF00FF
FillRectangle 100 100 100 100
```



FillText (code : hex 1A)

```
FillText baselineX baselineY text
```

Text 를 표시합니다. M 디스플레이는 < UTF-8 유니코드 > 를 지원합니다. 따라서 한글도 < UTF-8 유니코드 > 로 보내주어야 합니다.

EUC-KR (일명: 완성형코드)는 사용할 수 없습니다.

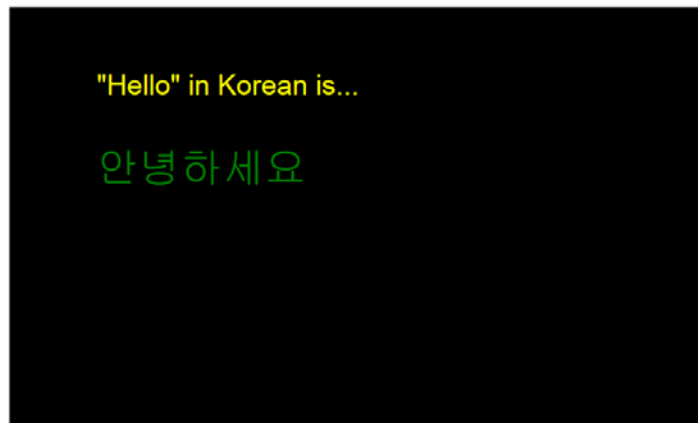
Argument	Type	Description
<i>baselineX</i>	Real Number	베이스라인의 X 좌표
<i>baselineY</i>	Real Number	베이스라인의 Y 좌표
<i>text</i>	Text	표시할 문자
<i>angle</i>	Real Number	(Optional) 문자의 회전각

```
Clear
```

```
# Draw yellow text in the default font
Color FFFF00
FillText 100 100 "\"Hello\" in Korean is..."

# Load font with Korean glyphs
LoadFont "/gulim.ttf"

# Draw green text in the gulim font
Color 00FF00
FontSize 48
FillText 100 200 "안녕하세요"
```



같은 위치에서 계속 갱신되는 숫자를 표시하는 경우에는 고려해야될 사항이 있습니다.
같은 위치에서 숫자를 다시 표시하면 아래 처럼 겹쳐져서 표시됩니다.

```
# Green Rectangle
Color 000800
FillRectangle 0 0 200 75

# White Text
Color FFFFFFFF
FillText 50 50 "123"
FillText 50 50 "456"
```



이를 피하기 위해, 해당 영역을 Clear 한뒤 표시한다면, 검은색 바탕이 보이게 됩니다.

```
# Green Rectangle
Color 000800
FillRectangle 0 0 200 75

# White Text
Color FFFFFFFF

# Display "123"
FillText 50 50 "123"

# Clear the box containing the text (black on background layer)
Clear 40 20 70 40

# Display "456"
FillText 50 50 "456"
```



Layer 기능을 이용하면 이 문제를 해결할 수 있습니다. 배경 Layer 에는 지워서는 안될 배경화면을 그리고, 앞 Layer 에 Text 를 표시합니다. 앞 Layer 를 Clear 한뒤 다음 숫자를 표시하면, 화면에는 자연스럽게 갱신되는 숫자가 표시됩니다.

```
# Green Rectangle
Color 000800
FillRectangle 0 0 800 480

# New 70x40 layer, positioned at 40,20 to contain text
CreateLayer 40 20 70 40 1
Layer 1

# White text
Color FFFFFFFF

# Display "123"
# Coordinates are relative to new layer's top-left corner (i.e. 40,20)
FillText 0 40 "123"

# Erase "123" (Transparent for foreground layer)
Clear

# Display "456"
FillText 0 40 "456"
```



456

FontSize (code : hex 1B)

FontSize size

폰트 사이즈를 결정합니다. 파워온시 디폴트값은 32 입니다.

Argument	Type	Description
<i>size</i>	Real Number	폰트 사이즈 (픽셀 단위)

```
# Clear the active layer
Clear

# Yellow text at default font size
Color FFFF00
DrawText 100 100 "Default Font Size"

# Green text at font size of 48
Color 00FF00
FontSize 48
DrawText 100 200 "Font Size 48"
```



GlossyButton (code : hex 1C)

```
GlossyButton topLeftX topLeftY width height cornerRadius buttonColor borderColor [text]
```

고급 버튼을 표시합니다.

Argument	Type	Description
<i>topLeftX</i>	Real Number	시작 X 좌표
<i>topLeftY</i>	Real Number	시작 Y 좌표
<i>width</i>	Real Number	폭
<i>height</i>	Real Number	높이
<i>cornerRadius</i>	Real Number	코너 반지름
<i>buttonColor</i>	Color	버튼의 색
<i>borderColor</i>	Color	테두리 색
<i>text</i>	Text	(Optional) 버튼 정중앙에 표시할 문자

```
# Clear the active layer
Clear

# White text
Color FFFFFFFF

# Border width of 2
LineWidth 2

# Draw a blue glossy button
GlossyButton 100 100 200 50 10 336699 336699 "Text"
```

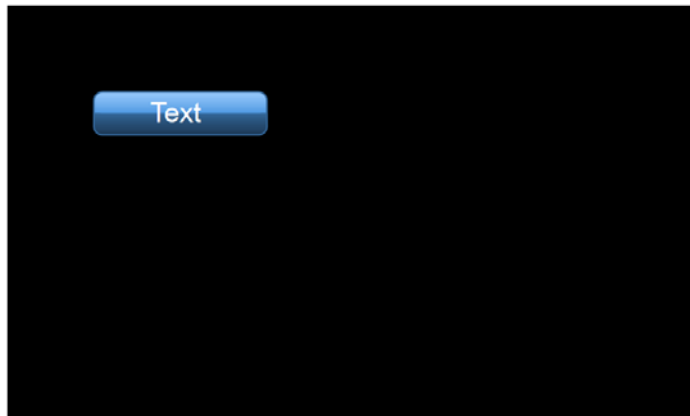


Image (code : hex 1D)

```
Image x y filePath
```

PNG 파일을 SD 카드로부터 읽어와 화면에 표시합니다.

Argument	Type	Description
<i>x</i>	Int16	왼쪽 상단 X 좌표
<i>y</i>	Int16	왼쪽 상단 Y 좌표
<i>filePath</i>	Text	경로명 (SD card)

```
# Clear the active layer
Clear

# Draw the image
Image 229 115 "/gauge.png"
```



Layer (code : hex 1E)

Layer layerID

Active Layer 를 정합니다. Background Layer 를 Active 하려면 Layer 0 을 사용하세요.

Argument	Type	Description
<i>layerID</i>	UInt8	Active 시킬 Layer 의 ID. 0 을 적으면 background Layer 가 Active 됩니다.

LineCap (code : hex 1F)

LineCap cap

선 마무리를 설정합니다.

Argument	Type	Description
<i>cap</i>	Text (UTF-8 protocol) UInt8 (binary protocol)	UTF-8 protocol 에서는 "Butt", "Round", "Square" 중 하나. Binary protocol 에서는 0, 1, 2 중 하나 (각각 Butt, Round, Square).

```
# Clear the active layer
Clear

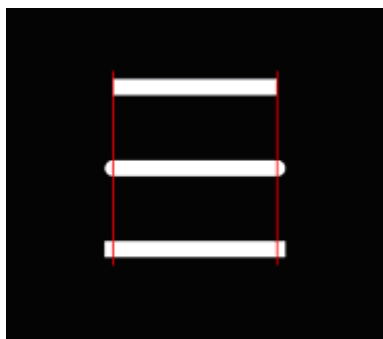
# Draw white lines with a thickness of 10
Color FFFFFFFF
LineWidth 10

# Draw three lines, each with a different end cap
LineCap Butt
DrawLine 100 100 200 100

LineCap Round
DrawLine 100 150 200 150

LineCap Square
DrawLine 100 200 200 200

# Draw red lines to illustrate the endpoints
LineCap Butt
LineWidth 1
Color FF0000
DrawLine 100 90 100 210
DrawLine 200 90 200 210
```



LineJoin (code : hex 20)

LineJoin join

선 접합 부위 표시방법을 설정합니다.

Argument	Type	Description
<i>join</i>	Text(UTF-8 protocol) UInt8 (Binary protocol)	UTF-8 protocol 에서는 "Miter", "Round", "Bevel" 중 하나. Binary protocol 에서는 0, 1, 2 중 하나 (각각 Miter, Round Bevel)

```
# Clear the active layer
Clear

# Set the line width
LineWidth 10.0

# Draw polyline with round join
ClearPath
MoveTo 100 100
LineTo 150 50
LineTo 200 100
LineJoin Round
DrawPath

# Draw polyline with miter join
ClearPath
MoveTo 250 100
LineTo 300 50
LineTo 350 100
LineJoin Miter
DrawPath

# Draw polyline with round join
ClearPath
MoveTo 400 100
LineTo 450 50
LineTo 500 100
LineJoin Bevel
DrawPath
```



LineTo (code : hex 21)

```
LineTo x y
```

현재 Point 로부터 지정된 위치까지 Path 를 생성합니다.

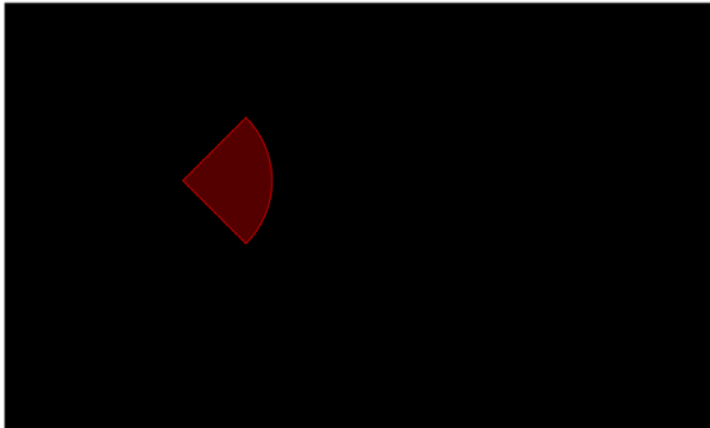
Argument	Type	Description
x	Real Number	X 좌표
y	Real Number	Y 좌표

```
# Clear the active layer
Clear

# Create a fraction of a pie
MoveTo 200 200
LineTo 270.7 129.29
ArcTo 270.7 270.7 100 100 0 0 1
ClosePath

# Fill the pie with partially transparent red
Color AAFF0000
FillPath

# Draw a border around the pie in bright red
LineWidth 1
Color FF0000
DrawPath
```



LineWidth (code : hex 22)

LineWidth width

라인 굵기를 설정합니다.

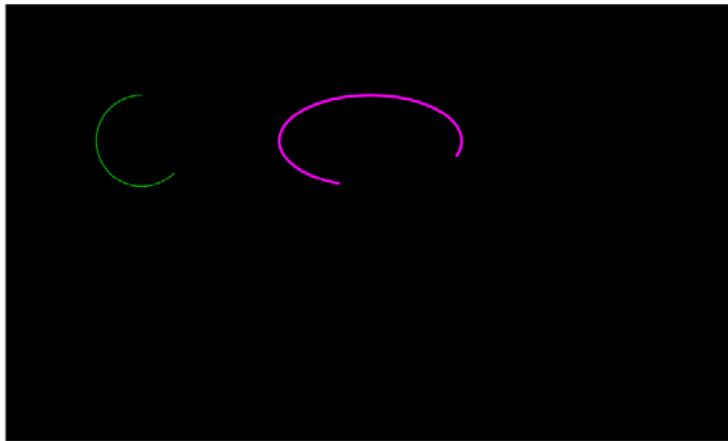
Argument	Type	Description
<i>width</i>	Real Number	굵기 (픽셀단위)

```
# Clear the active layer
Clear

# Draw green circular arc with a stroke width of 1.0
LineWidth 1.0
Color 00FF00

# Width omitted for circular arc
DrawArc 100 100 50 45.0 235.0

# Draw fuchsia elliptical arc with a stroke width of 3.0
LineWidth 3.0
Color FF00FF
DrawArc 300 100 100 50 110 270
```



LinearGradient (code : hex 23)

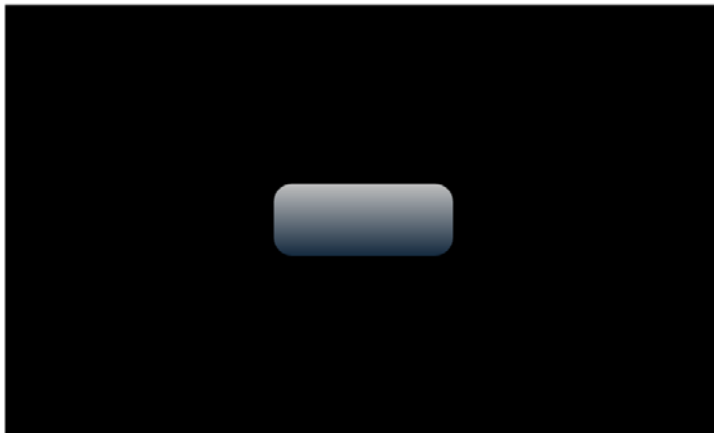
```
LinearGradient startX startY startColor endX endY endColor
```

그라데이션 칼라 패턴을 설정합니다.

Argument	Type	Description
<i>startX</i>	Real Number	시작 X 좌표.
<i>startY</i>	Real Number	시작 Y 좌표.
<i>startColor</i>	Color	시작 칼라
<i>endX</i>	Real Number	끝 X 좌표.
<i>endY</i>	Real Number	끝 Y 좌표
<i>endColor</i>	Color	종료 칼라

```
# Clear the active layer
Clear

# Draw a rounded rectangle with a linear gradient fill
LinearGradient 400 200 C0C0C0 400 280 336699
FillRoundedRectangle 300 200 200 80 20
```



LoadFont (code : hex 24)

```
LoadFont filePath [cacheInMemory]
```

SD 카드에 있는 폰트파일 (ttf 확장자)중 하나를 선택합니다.

Argument	Type	Description
<i>filePath</i>	Text	SD 카드의 파일경로
<i>cacheInMemory</i>	UInt8	(optional) 생략시 0 0 = SD 카드에서 바로 읽어옵니다. 1 = SD 카드에서 내부 메모리로 불러들인 뒤 읽어옵니다. 0 으로 하면 항상 SD 카드에서 읽어오므로 내부메모리를 차지하지 않습니다. 단, 표시속도가 느립니다. 따라서, 자주 사용하는 폰트는 1 로 하는 것이 좋습니다. 단 내부메모리에 들어갈 수 있는 크기의 폰트파일이어야 합니다. 너무 큰 폰트파일은 읽어올수 없습니다. Subsetfont 유틸리티를 이용해서 작은크기의 폰트파일로 변경하여 사용하시는 것이 좋습니다. <부록>에 있는 <폰트 서브셋 유틸리티> 설명 참조.

```
# Clear the active layer
Clear

# Display font in white
Color FFFFFFFF

#Load Font (stream from disk)
LoadFont "/font.ttf"
FillText 100 100 "This is text"

#Load Font (cache in memory)
LoadFont "/fonti.ttf" 1
Filltext 100 200 "This is Italic text"
```



MoveLayer (code : hex 25)

```
MoveLayer topLeftX topLeftY
```

현재 Active 된 Layer 를 재배치합니다. Background Layer 는 움직일 수 없습니다.

Argument	Type	Description
<i>topLeftX</i>	Int16	움직일 곳의 좌측상단 X 좌표
<i>topLeftY</i>	Int16	움직일 곳의 좌측상단 Y 좌표

```
# Clear the active layer
Clear

# Create layer 1 at (100, 100)
CreateLayer 100 100 200 200 1

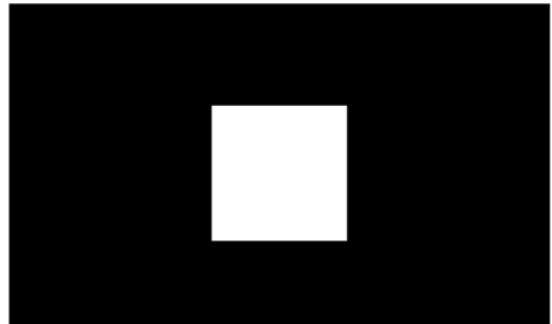
# Change the active layer to layer 1
Layer 1

# Fill a rectangle that covers the entire layer
FillRectangle 0 0 200 200

# Move the entire layer to (300, 150)
MoveLayer 300 150
```



실행전



실행후

MoveTo (code : hex 26)

```
MoveTo x y
```

현재 Point 를 x, y 좌표로 이동합니다.

Argument	Type	Description
x	Real Number	움직일 곳의 X 좌표
y	Real Number	움직일 곳의 Y 좌표

Ping (code : hex 27)

Ping

응답 메시지를 보냅니다. 통신 상태를 체크할 목적으로 사용합니다.

RadialGradient (code : hex 28)

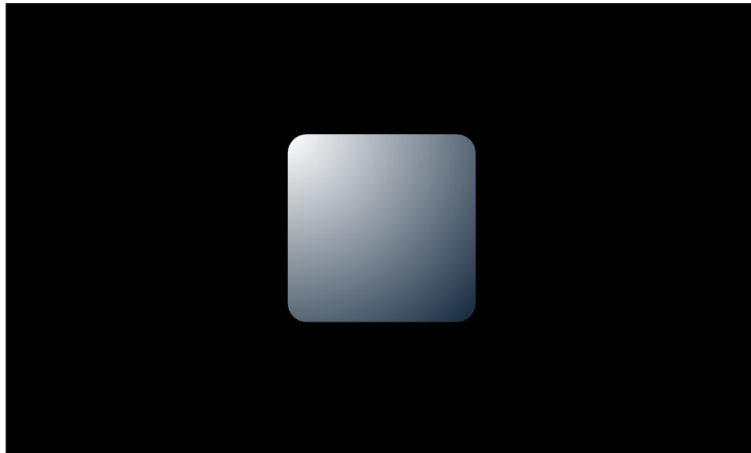
```
RadialGradient startX startY startColor radius endColor
```

그라데이션 칼라패턴을 설정합니다.

Argument	Type	Description
<i>startX</i>	Real Number	시작 X 좌표
<i>startY</i>	Real Number	시작 Y 좌표
<i>startColor</i>	Color	시작 칼라
<i>radius</i>	Real Number	시작지점으로부터 반지름
<i>endColor</i>	Color	종료 칼라

```
# Clear the active layer
Clear

# Draw a rounded rectangle with a radial gradient fill
RadialGradient 300 140 C0C0C0 280 336699
FillRoundedRectangle 300 140 200 20 20
```



RunScript (code : hex 29)

```
RunScript filePath
```

SD 카드로부터 Script 파일을 읽어서 실행합니다. Script 파일이란 M Display의 명령어가 기록된 Text 파일입니다. PC의 메모장 (Notepad)와 같은 Text Editor로 작성하신뒤 SD 카드로 Copy 하신뒤 사용하십시오.

- Script 파일내에서 # 이후 문장은 무시합니다. 코멘트를 적는 용도로 사용하십시오.
- UTF-8 명령포맷을 사용하여 작성하십시오. (Binary 프로토콜을 사용하고 있을때에도)
- 한 라인에 하나의 명령문만 사용하십시오.
- Script 파일내에서 RunScript 명령을 사용할 수 없습니다.
- Script 파일내의 각각의 명령에 대해 응답메시지가 송신되지 않고, 모든 명령문 실행이 정상적으로 끝나면 성공 응답메시지가 단 한번 송신됩니다.

Argument	Type	Description
<i>filePath</i>	Text	SD 카드의 파일경로

사용예)

“/Scripts/Grid.txt”에 다음과 같은 내용의 Script를 저장합니다.

```
# Clear the active layer
Clear

# Don't draw anything until all commands have executed
DisableFlush

# Set line width
LineWidth 2.0

# White lines
Color FFFFFFFF

# Vertical lines
DrawLine 10 10 10 470
DrawLine 30 10 30 470
DrawLine 50 10 50 470
DrawLine 70 10 70 470
DrawLine 90 10 90 470
DrawLine 110 10 110 470
DrawLine 130 10 130 470
DrawLine 150 10 150 470
```



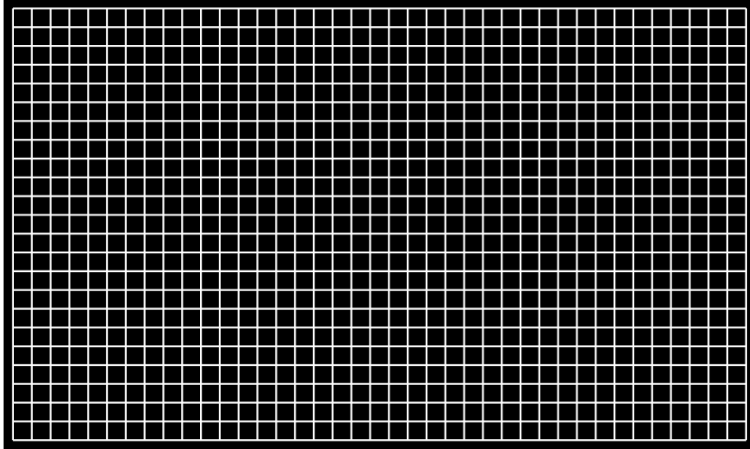
```
DrawLine 170 10 170 470
DrawLine 190 10 190 470
DrawLine 210 10 210 470
DrawLine 230 10 230 470
DrawLine 250 10 250 470
DrawLine 270 10 270 470
DrawLine 290 10 290 470
DrawLine 310 10 310 470
DrawLine 330 10 330 470
DrawLine 350 10 350 470
DrawLine 370 10 370 470
DrawLine 390 10 390 470
DrawLine 410 10 410 470
DrawLine 430 10 430 470
DrawLine 450 10 450 470
DrawLine 470 10 470 470
DrawLine 490 10 490 470
DrawLine 510 10 510 470
DrawLine 530 10 530 470
DrawLine 550 10 550 470
DrawLine 570 10 570 470
DrawLine 590 10 590 470
DrawLine 610 10 610 470
DrawLine 630 10 630 470
DrawLine 650 10 650 470
DrawLine 670 10 670 470
DrawLine 690 10 690 470
DrawLine 710 10 710 470
DrawLine 730 10 730 470
DrawLine 750 10 750 470
DrawLine 770 10 770 470
DrawLine 790 10 790 470

# Horizontal lines
DrawLine 10 10 790 10
DrawLine 10 30 790 30
DrawLine 10 50 790 50
DrawLine 10 70 790 70
DrawLine 10 90 790 90
DrawLine 10 110 790 110
DrawLine 10 130 790 130
DrawLine 10 150 790 150
DrawLine 10 170 790 170
DrawLine 10 190 790 190
DrawLine 10 210 790 210
DrawLine 10 230 790 230
DrawLine 10 250 790 250
DrawLine 10 270 790 270
DrawLine 10 290 790 290
DrawLine 10 310 790 310
DrawLine 10 330 790 330
DrawLine 10 350 790 350
DrawLine 10 370 790 370
DrawLine 10 390 790 390
DrawLine 10 410 790 410
DrawLine 10 430 790 430
DrawLine 10 450 790 450
DrawLine 10 470 790 470

# Display the result
EnableFlush
```

실행 결과는 다음과 같습니다.

```
RunScript "/Scripts/Grid.txt"
```



Reset (code : hex 2A)

Reset

MDisplay 를 리셋시킵니다.

SetPixel (code : hex 2C)

```
SetPixel x y color0 [color1 ... colorn]
```

X, Y로부터 수평으로 점을 그립니다. Argument 개수에 따라 다수의 점을 연속으로 그릴 수 있습니다.

Argument	Type	Description
<i>x</i>	Int16	X 좌표
<i>y</i>	Int16	Y 좌표
<i>color0 ...colorn</i>	Color	수평으로 배열될 칼라 리스트 (<i>color0</i> at <i>x,y</i> ~ <i>colorn</i> at <i>x+n,y</i>).

```
# Clear the active layer
Clear

# First row. Even pixels red.
SetPixel 100 100 FF0000 000000 FF0000 000000 FF0000 000000 FF0000

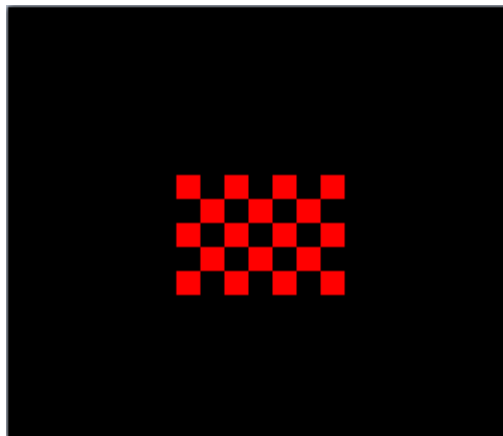
# Second row. Odd pixels red.
SetPixel 101 100 FF0000 000000 FF0000 000000 FF0000

# Third row. Even pixels red.
SetPixel 100 100 FF0000 000000 FF0000 000000 FF0000 000000 FF0000

# Fourth row. Odd pixels red.
SetPixel 101 100 FF0000 000000 FF0000 000000 FF0000

# Fifth row. Even pixels red.
SetPixel 100 100 FF0000 000000 FF0000 000000 FF0000 000000 FF0000
```

다음은 해당부분을 확대한 사진입니다.



UnloadFont (code : hex 2D)

UnloadFont

현재 선택되어 있는 Font 를 선택해지합니다.

제 4 장

라이브러리

MOACON 용 라이브러리

M Display 과 Moacon 을 사용하시는 분들을 위한 <라이브러리>입니다. 모아콘을 사용하시는 분들은 본 라이브러리를 사용하여 마치 명령어를 사용하듯 편리하게 M display 를 이용하실 수 있습니다.

라이브러리는 이곳에서 다운로드 받으실 수 있습니다.

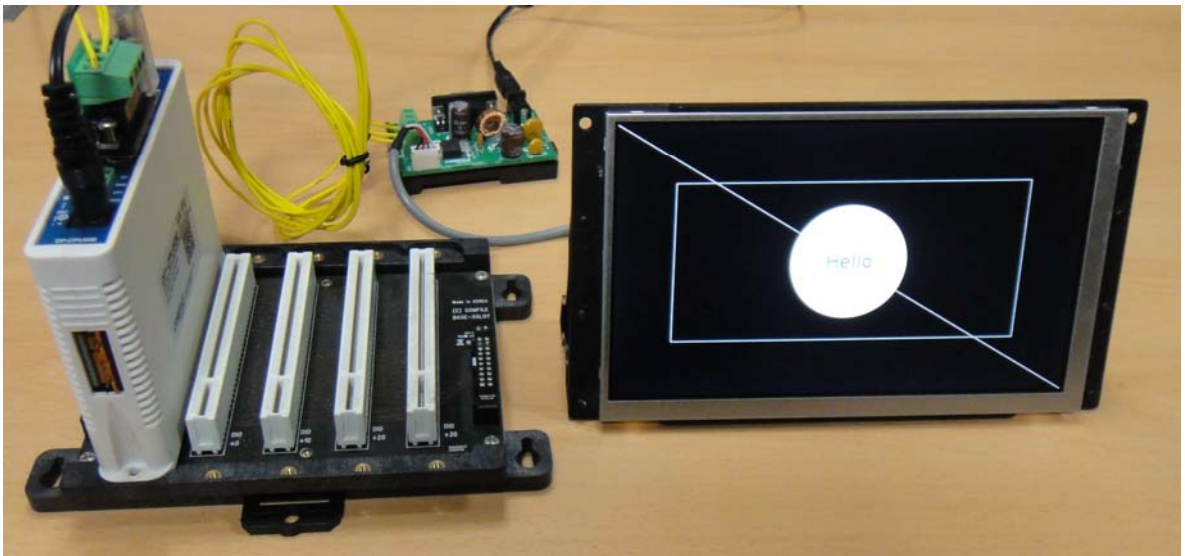
http://www.comfile.co.kr/download/display/Moacon_MDisplayLibrary.zip

M Display 는 파워온시 디폴트로 다음과 같은 상태가 됩니다.

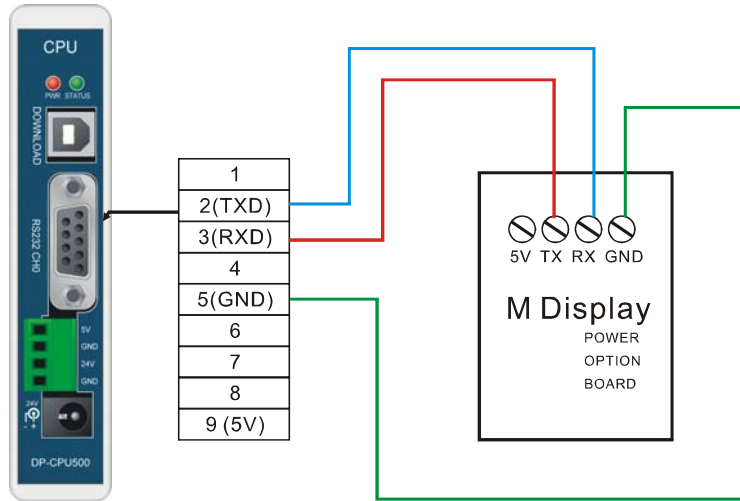
종류	파워온시 상태
통신 모드	115200 보레이트, 8 비트, No 패리티, 1 스톱비트
프로토콜	UTF-8
체크섬	사용안함

이 상태 그대로, 아래 소스 프로그램을 사용하시면, M display 화면에 그림이 표시됩니다.

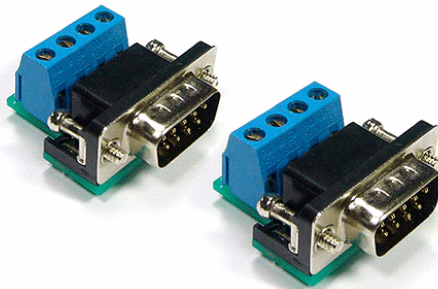
아래 사진에서 처럼, 모아콘의 채널 0 과 Mdisplay 를 서로 연결하였습니다.



결선은 다음과 같이 하십시오.



모아콘에 <RS232 터미널 블록 콘택>을 사용한 것입니다.



다음은 라이브러리를 사용하는 메인 프로그램입니다. (파일명 : Test.c)

```
#include "moacon500.h"
#include <math.h>

#include "mdisplay.h"

static void MD_OnError(MD_Status status)
{
    printf("M Display: Error - 0x%04X\r\n", status);
}

void cmain ()
{
    //Uncomment to change timeout or protocol
    //-----
    // MD_SetTimeout(1000);
    // MD_SetProtocol(MD_Protocol_Binary);
    // MD_EnableChecksum();

    MD_SetOnError(MD_OnError);
    MD_OpenCom(0, 115200, com8N1);

    while(1)
    {
        MD_DisableFlush();
        MD_Clear();
        MD_Color(0xFFFFFFFF);
        MD_DrawLine(0, 0, 800, 480);
        MD_DrawRectangle(100, 100, 600, 280);
        MD_FillCircle(400, 240, 100);
        MD_Color(0x0000FF);
        MD_FillText(360, 250, "Hello");
        MD_EnableFlush();

        MD_DisableFlush();
        MD_Clear();
        MD_Color(0x00FF00);
        MD_DrawLine(800, 0, 0, 480);
        MD_DrawRectangle1(100, 100, 600, 280, 20);
        MD_FillEllipse(400, 240, 150, 100);
        MD_Color(0xFF00FF);
        MD_FillText(360, 250, "Hello");
        MD_EnableFlush();
    }
}
```

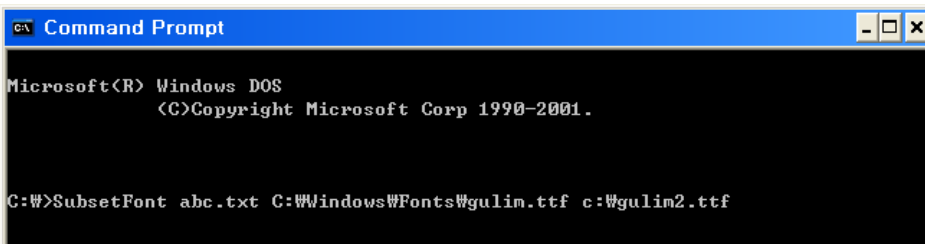
부록

폰트 서브셋 유틸리티

일반적으로 한글 ttf 파일은 용량이 커서, LoadFont 명령 수행시 로드 시간이 소요됩니다. 좀 더 빠른 표시를 위해서 “SubsetFont” 유틸리티를 제공합니다.

(홈페이지 M 디스플레이 상세페이지에 Download 탭에 있는 유틸리티 모음에 들어있습니다.)

한글 ttf 파일에는 < 벗, 흘, 꿇, 춤 > 과 같은 잘 쓰이지 않는 한글과 한자까지 모두 포함하고 있기 때문에 용량이 매우 큽니다. 이중 실제 사용하는 폰트만 추출해서 작은 사이즈의 ttf 파일을 만들어 주는 유틸리티입니다. “SubsetFont” 프로그램은 command Promt 창에서 실행합니다.



```

CA Command Prompt
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-2001.

C:\>SubsetFont abc.txt C:\Windows\Fonts\Gulim.ttf c:\Gulim2.ttf
  
```

메모장을 사용해서 Abc.txt 파일을 만들어 주십시오. 이 안에는 여러분의 어플리케이션에서 사용하는 문자만 적어주십시오.

컴파일테크놀로지메인메뉴설정초기화가스주입추출재생시작종료안녕하세요

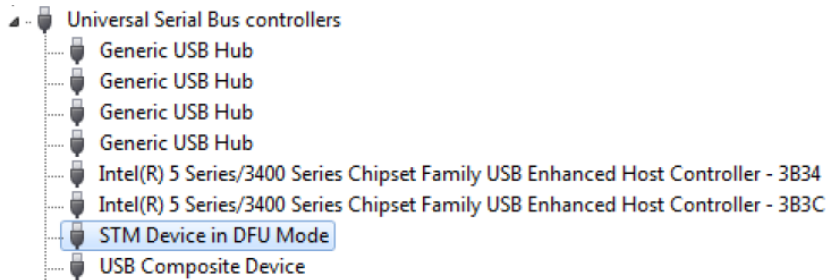
이렇게 한다면 기존 gulim.ttf 에서 원하는 캐릭터만 추출해서 gulim2.ttf 를 생성합니다.

이 ttf 파일을 SD 카드로 옮긴뒤, M Display 에서 loadFont 명령에서 사용하면 폰트 로드시간이 단축됩니다.

펌웨어 업그레이드

M Display 에 새로운 펌웨어를 다운로드할 때 쓰는 유틸리티 (DFU)는 www.comfile.co.kr 에서 다운로드받으실 수 있습니다.

1. 덤스위치 2 번을 on 위치로 바꾸고, M display 의 전원을 On 합니다.
2. PC 와 USB 케이블로 연결하십시오.
3. PC 에서 Driver Install Wizard 가 실행되면, DFU 유틸리티 Driver 폴더를 지정하십시오.
4. USB 드라이버가 성공적으로 설치되면, PC 의 <시스템등록정보> <장치관리자>에서 다음과 같은 상황을 볼 수 있습니다. (STM Device in DFU mode 가 새롭게 설치됩니다.)



DFU 는 command prompt 실행창에서 실행시키십시오.

M display 로 다운로드할 최신 펌웨어 (hex 파일)은 www.comfile.co.kr 에서 다운로드 받으실 수 있습니다. 명령어는 다음과 같습니다. 해당 실행파일이 있는 폴더로 이동하셔서 실행하셔야 합니다.

```
MdisplayDFU Mdisply070_v0.9.9.hex
```

```

C:\ Command Prompt
2013-09-09 오전 11:21 <DIR> M Display Console v1.0.0.0
2013-09-09 오전 11:24 <DIR> M Display DFU v1.0.0.0
2013-09-09 오전 11:21 <DIR> M Display Hardware Version Writer v1.0.0.0
2013-09-09 오전 11:21 <DIR> M Display Test Program v1.0.0.0
2013-09-09 오전 11:18 2,132,544 MDisplay070_v0.9.9.hex
2013-08-27 오전 02:09 23,552 MDisplayDFU.exe
2013-08-26 오전 03:10 15,872 ST.dll
2013-08-27 오전 01:29 53,248 STDFU.dll
2013-08-27 오전 01:29 257,536 STDFUFiles.dll
2013-08-27 오전 01:29 224,768 STDFUPRT.dll
2013-08-27 오전 01:29 225,280 STTubeDevice30.dll
2013-09-09 오전 11:21 <DIR> SubsetFont 1.0.0.0
       7개 파일 2,932,800 바이트
       7개 디렉터리 753,557,254.144 바이트 남음

D:\WMDISP_~1>MDisplayDFU MDisplay070_v0.9.9.hex
MDISPLAYDFU v1.0.0.0
Downloading "MDisplay070_v0.9.9.hex" <758144 bytes>
Erasing: 100%
Downloading: 100%

D:\WMDISP_~1>
    
```

펌웨어 다운로드가 모두 끝나면, Dip 스위치 2 번을 다시 Off 위치로 바꾸고 사용하십시오.

알림: M Display 성능 개선을 위해서, 새로운 펌웨어는 사전 예고 나 개별 연락없이 홈페이지 (www.comfile.co.kr)자료실에 공지됩니다.

M Display 콘솔 프로그램

M Display의 실행상태를 모니터링 할 수 프로그램입니다. 이를 위해서 M Display의 USB 포트와 PC가 서로 연결되어 있어야 합니다.

www.comfile.co.kr에서 M 디스플레이 제품 상세페이지의 Download 탭의 유틸리티모음안에 콘솔 프로그램이 있습니다.



 [M DISPLAY 사용설명서]

 [M DISPLAY 디맨전]

 [유틸리티 모음]

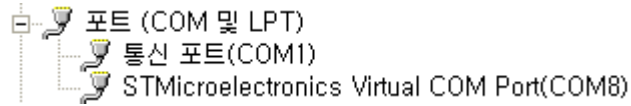
 [모아콘용 라이브러리]

1. Dip 스위치 2 번이 Off 상태 (Run 모드)에서 M Display의 전원을 On 하세요.
2. PC 화면에 <새 하드웨어 검색 마법사>가 표시되지만 무시하시고, 다운로드받은 콘솔 프로그램의 Driver 폴더로 가서 VCP_Vx.x.x_Setup 프로그램을 실행시키십시오.
(64 비트 버전 윈도우를 쓰고 있다면 VCP_Vx.x.x_Setup_x64 를 실행시키십시오.)



3. 설치가 완료되면 <하드웨어 검색 마법사>가 한번 더 표시됩니다. <다음>을 누르시면 최종적으로 완료 화면이 표시됩니다.

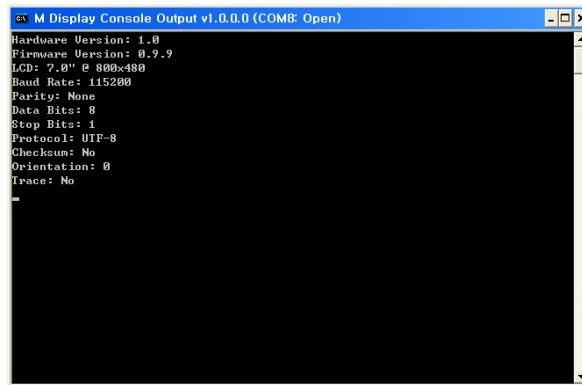
4. <장치관리자>에서 Virtual COM port 가 설치되어 있는 것을 확인하시면, 모든 설치가 정상적으로 종료된 것입니다.



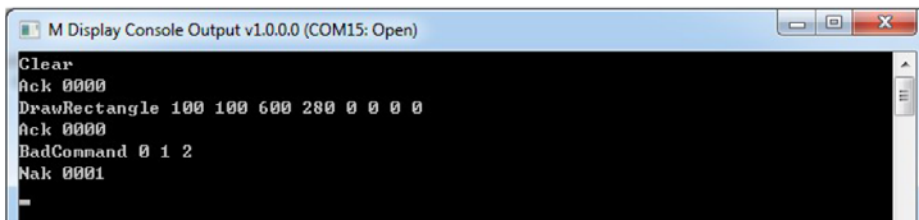
이제 MdisplayConsole 프로그램을 실행시킨후, M display 의 전원을 On 하십시오.



아래와 같은 셋팅정보가 표시됩니다.



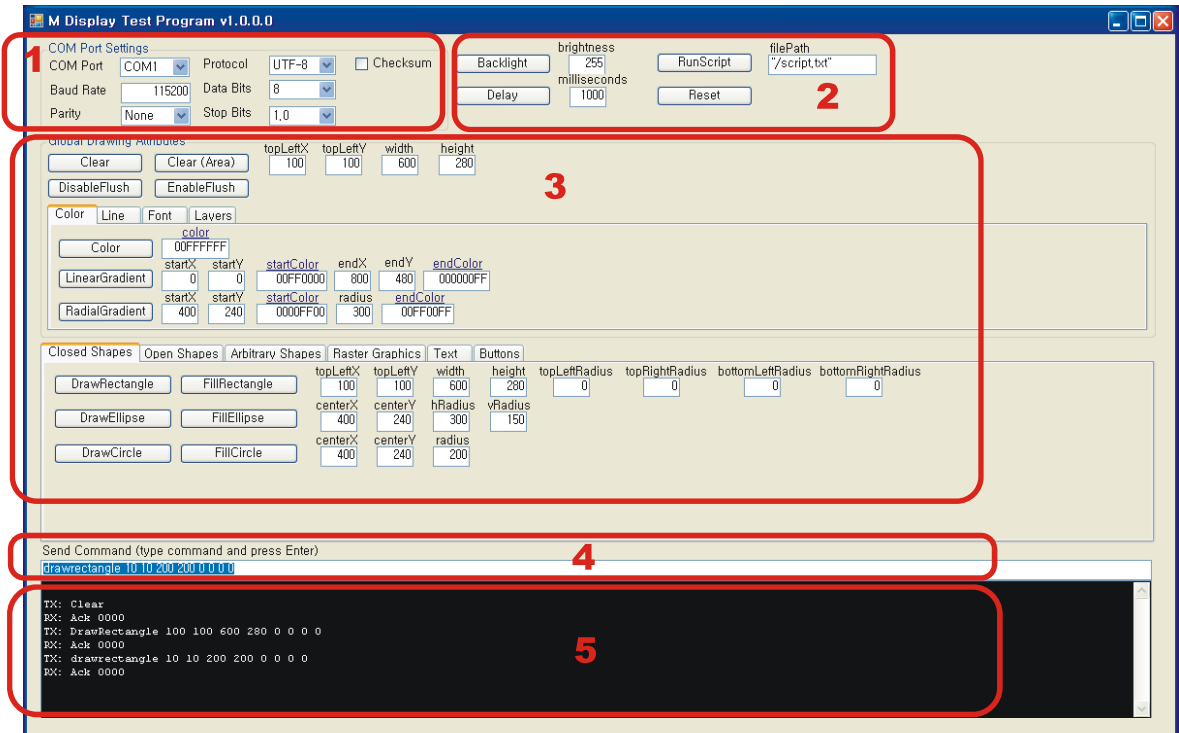
Init.txt 에 EnableTrace 명령이 있다면, 아래처럼 모든 명령어의 실행상황이 콘솔프로그램안에 표시됩니다.



M Display Test 프로그램

M Display 를 테스트 해볼 수 있는 프로그램입니다. 앞에서 받은 유틸리티 안에 들어 있습니다. PC 와 M display 를 서로 연결한뒤 사용하십시오.

이 프로그램은 .net framework 4 이상이 설치되어 있어야 동작됩니다. (microsoft 공식사이트에서 다운로드 가능)



1. 통신설정 : PC 의 통신포트 및 프로토콜 상태를 입력하세요.
2. 보조 코멘드
3. 그래픽 코멘드 : <UTF-8>프로토콜에서만 사용가능합니다.
4. 직접 명령 실행창
5. 결과 표시 창