LN2440SBC 임베디드 시스템을 위한 TFT LCD 초기화 및 그래픽스 라이브러리 함수 구현

김병국*, 박근덕*, 오삼권*
*호서대학교 컴퓨터공학과
e-mail:greatkuky@naver.com

The Initialization of a TFT LCD and Implementation of Library Functions for an LN2440SBC Embedded System

Byoung Kuk Kim*, Geun Duk Park*, Sam Kweon Oh*
*Dept. of Computer Engineering, Hoseo University

요 약

LN2440SBC 임베디드 보드는 ARM 코어 방식의 S3C2440A CPU를 가진 임베디드 컴퓨터 시스템이다. 이 시스템에 부착한 터치스크린 기능을 가진 TFT LCD 키트인 LP35의 구동을 위해서는 ARM 코어, LCD 컨트롤러, 그리고 LCD 장치와의 통신을 위한 SPI(serial peripheral interface)의 초기화와 LCD 화면에 이미지, 선, 도형 같은 것들의 출력을 가능하게 해주는 그래픽스 라이브러리 함수들이 필요하다. 본 논문은 이같은 기능들을 가지는 LP35를 위한 드라이버의 구현 방법을 기술한다. 특히, 드라이버 구동을 위한 초기화 방법과 화면 출력 기능들의 구현을 위해 필요한 픽셀 디스플레이 함수의 구현에 중점을 두어 설명한다. 또한 픽셀 디스플레이 함수를 이용한 기본 그래픽스 라이브러리 함수들에 대해 설명한다. 드라이버의 초기화를 위해서는 클럭 속도 설정, 범용 입출력 핀(GPIO)을 LCD와 SPI 용으로의 할당. SPI의 마스터/슬레이브 및 보오 레이트 설정, LCD 컨트롤러 레지스터 설정을 통한 LCD 기능 선택. 그리고 SPI를 통한 LCD 장치로의 파워 온(power on) 명령 전달 등이 수행된다.

1. 서론

장치 드라이버(device driver)는 하드웨어와 응용 프로그램 사이의 매개체 역할을 하는 핵심 구성 요소로서, 응용 프로그램이 하드웨어를 제어하여 상호동작하기 위한 일관된 인터페이스를 제공하는 소프트웨어이다. 즉 실질적인 하드웨어 제어는 장치 드라이버가 수행하고 응용프로그램은 장치 드라이버가 제공하는 인터페이스를 사용하여 해당 장치의 기능들을 실행함으로서 프로그래밍을 용이하게 해준다. 일반적으로 장치 제조업체가 해당 드라이버를 제공하는 PC와는 달리, 임베디드 시스템에서는 직접 사용목적에 맞는 전용 드라이버를 제작하여 사용해야 하며 이런 전용 장치 드라이버의 제작은 해당 하드웨어와 소프트웨어의 깊은 이해를 요구한다.

본 논문은 시랩시스(CLabSys)사의 LN2440SBC 임베디드 보드(S3C2440A CPU, 400MHz)에 터치스크린 기능 있는 3.5″ TFT LCD 키트인 LP35(3.5" TFT LCD Kit with Touch Screen)를 부착하고 이 LP35 TFT LCD 드라이버의 제작을 위한 전반적인 초기화 방법과 픽셀 디스플레이 방법을 기술 한다. 또한 사용자와의 인터페이스나 데이터의 가시화를 위한 이미지, 선, 도형들을 그리는 그래픽스 라이브러리 함수에 대해서 설명한다.

본 논문의 구성은 다음과 같다. 2장은 3.5“ TFT LCD 키트인 LP35와 LN2440SBC 임베디드 보드, S3C2440A LCD 컨트롤러를 설명한다. 3장은 TFT LCD 구동을 위한 초기화 방법을 설명하고 4장은 TFT LCD 디스플레이를 위한 픽셀 디스플레이 함수와 그래픽스 라이브러리 함수의 구현을 설명한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

LCD는 수동 매트릭스 방식(passive matrix type)과 능동 매트릭스 방식(active matrix type)으로 구분된다. 수동 매트릭스 방식의 LCD로는 TN LCD와 STN LCD가 있는데, 이들은 좁은 시야각과 저속 재생율 등의 문제점을 가지고 있다[1]. 이 문제들의 해소를 위해 능동 매트릭스 방식이 제안되었으며 현재 가장 널리 보급되어 있는 TFT LCD는 이 방식을 채택하고 있다.

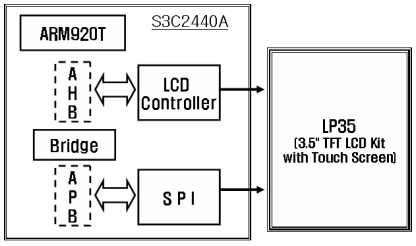
삼성의 TFT LCD인 LTV350QV-F04 LCD[2]가 내장된 LP35는 320*240의 크기를 가지고, 24 비트의 컬러를 사용하며, 최대 16M 컬러를 지원하고, LCD 제어 명령의 수신을 위해 SPI를 사용한다. LCD 구동을 위해 픽셀 클럭 신호와 데이터, 데이터 Enable 신호 및 VCOM 신호등을 사용하는 비동기식의 DE모드와 픽셀 클럭과 데이터, VSYNC, HSYNC, VCOM등을 사용하는 동기식의 SYNC 모드를 지원한다. 이 중 LP35는 SYNC 모드로 구성되어

있다.

LN2440SBC 임베디드 보드[3]는 (그림 1)과 같은 구조를 가진다. 이 보드에 내장된 S3C2440A 마이크로프로세서[1][4]에는 ARM920T 코어와 TFT/STN LCD 컨트롤러, SPI 등이 내장되어 있고 터치스크린 기능을 가진 3.5″ TFT LCD 키트인 LP35가 부착되어 있다.

ARM920T와 LCD 컨트롤러는 고속으로 동작하는 장치들을 위한 AHB(advanced high performance bus) 버스로 통신을 하고 SPI와는 저속의 주변장치를 위한 APB(advanced peripheral bus) 버스로 통신을 한다. SPI[5]는 프로세서에 내장된 동기식 직렬 전송을 지원하는 전이중 방식의 직렬 인터페이스이며, 통신하는 한 쌍의 장치중 하나에 의해 생성된 동일한 클럭 신호에 의해 동기화 되는데, 이 경우 마스터-슬레이브 관계가 형성된다. 입출력 핀으로 SPIMOSI(master out, slave in pin)와 SPIMISO(master in, slave out pin)가 있으며, SPI가 마스터 모드일 경우 SPIMOSI는 출력 핀, SPIMISO는 입력 핀이 된다. ARM920T는 SPI를 통해 LCD 장치로 제어 명령을 전송하며 LCD 컨트롤러는 GPIO(general purpose input output)를 통해 LCD 제어신호와 데이터를 LCD 장치로 전송한다. GPIO[1][5]는 범용 입출력 장치 핀으로써 리셋이나 그라운드, 클럭 핀과 같이 특수한 목적으로 할당된 것이 아니라 프로그램에서 어떻게 설정하는가에 따라 기능을 여러 용도로 사용할 수 있는 핀을 말한다.

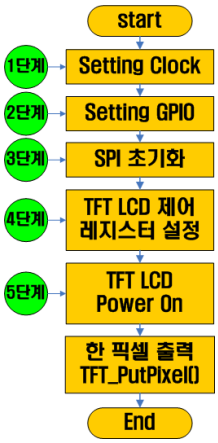
S3C2240A 프로세서에 내장된 S3C2440A의 LCD 컨트롤러[1][4]는 프레임버퍼와 외부 LCD 드라이버 사이에서 비디오 데이터 전송을 맡아서 처리하는 제어 로직이며, LCD 전용 DMA(direct memory acess) 컨트롤러는 프레임버퍼에 있는 비디오 데이터를 CPU의 중재 없이 LCD 드라이버로 전송 한다. S3C2440A LCD 컨트롤러는 단색 LCD부터 컬러 STN LCD, TFT LCD를 지원하며, TFT LCD의 경우 1, 2, 4, 8, 16, 24 bpp(bits per pixel) 컬러 디스플레이를 지원하고, 640x480, 320x240, 160x160 등의 다양한 스크린 크기를 지원한다. 가상 스크린 크기는 최대 4 MB를 지원한다.



(그림 1) LN2440SBC 블록 다이어그램

3. TFT LCD 구동을 위한 초기화

하드웨어는 TFT LCD 키트인 LP35와 ARM920T기반의 삼성 S3C2440A CPU가 내장된 LN2440SBC 임베디드 보드를 사용하고, 컴파일러는 ARM Developer Suite v1.2를 사용하며, 임베디드 보드로의 실행 이미지 다운로딩을 위해 시랩시스사의 Arm Down v3.8을 사용 한다.
(그림 2)는 TFT LCD의 구동을 위한 초기화 과정이다



(그림 2) TFT LCD 구동 순서도

1 단계에서는 각 장치로 전달되는 클럭 속도를 설정한다. S3C2440A 매뉴얼의 권장 값에 따라 코어에서 사용되는 FCLK 클럭은 405 MHz로 설정하고, 인터럽트 컨트롤러, DMA, LCD 컨트롤러 등의 고성능 장치에서 사용하는 HCLK 클럭은 67.5 MHz로, ADC, UART, GPIO, SPI 등의 저전력 주변장치에서 사용하는 PCLK 클럭은 33.75 MHz로 설정한다. S3C2440A LCD 컨트롤러가 사용하는 HCLK 클럭은 디스플레이 장치가 화면 하나의 데이터를 표시하는 속도인 프레임 레이트(frame rate)의 속도에 영향을 끼친다. 프레임 레이트는 평균 60 fps(frames per second)정도가 적당하며, HCLK가 67.5 MHz일 경우 약 62 fps의 프레임 레이트를 얻을 수 있다.

2단계에서는 GPIO를 설정한다. S3C2440A는 입출력 포트 8개(GPB~GPJ)와 출력전용 포트 1개(GPA)가 있으며 총 130핀의 GPIO를 가지고 있다. GPC 포트와 CPD 포트의 GPIO 핀들을 LCD 제어신호와 데이터 버스용으로 할당하여 LCD를 구동시킨다. 코어와 LCD 장치와의 통신 인터페이스인 SPI의 사용을 위해 GPE 포트의 핀들을 SPI 입출력 핀인 SPIMOSI, SPIMISO로 할당하고, SPI 클럭 핀으로 지정한다. SPI의 클럭으로는 PCLK 클럭을 사용한다.

3단계에서는 LCD 장치로 제어 명령을 전송하는 SPI를 초기화 한다. SPI를 마스터 모드로 설정하여 SPIMOSI는 출력 핀, SPIMISO는 입력 핀으로 지정하고, 전송 속도인 보오 레이트를 설정한다. 다음은 보오 레이트의 계산 수식이다:

보오 레이트 = PCLK / 2 / (프리스케일러 값 + 1)

클럭의 속도를 조절해주는 제수인 프리스케일러 값을 지정함으로써 보오 레이트를 설정할 수 있다. 단, 프리스케일러 값은 최대 255까지 지정할 수 있으며, 보오 레이트는 25 MHz 이하여야 한다. 보오 레이트의 최저 속도는 프리

스케일러 값을 최대값인 255로 지정하였을 경우로서 49 KHz이다.

4단계에서는 LCD 컨트롤러의 기본 설정을 담당하는 LCDCON1에서 LCDCON4까지의 레지스터들과 프레임 버퍼 주소 레지스터 등, LCD 컨트롤러를 제어하기 위한 레지스터들을 설정함으로서 TFT LCD의 기능을 설정 한다. LP35는 320*240 개의 픽셀(해상도)을 가지는 TFT LCD 이므로, 화면 크기를 수평 320과 수직 240으로 각각 설정하고 출력 해상도도 320*240으로 설정하며 디스플레이모드는 TFT LCD 패널 모드로 설정한다. 현재 가상 스크린은 사용하지 않기 때문에 크기를 320*240의 크기로 고정하며 LCD 제어 모드는 LP35에서 지원하는 Sync 모드로 설정한다. CLKVAL 값은 프레임 레이트와 픽셀 클럭에 영향을 주며, 60 fps 정도의 프레임 레이트를 얻기 위해 그 값을 5로 설정한다. 보통 영상 출력시 30 fps 이상이 필요하고, 부드러운 표시를 위해서는 60 fps~120 fps가 적당하다. LP35는 최대 24 bpp의 색상을 지원하지만 320*240의 작은 해상도이므로, 사람의 눈엔 16 bpp와 큰 차이가 없다 따라서 오버헤드를 줄이기 위해 16 bpp를 선택한다. 16 bpp 색상은 RGB 비율이 5:6:5인 형식과 5:5:5:I 형식을 지원하며. I는 밝기 조절에 이용된다. 본 논문에서는 밝기 조절을 제외한 5:6:5 형식을 사용하기로 한다. <표 1>은 본 논문에서 설정한 주요 항목의 값들이다. 상황에 따라 다른 설정 값을 선택할 수 있으며 자세한 내용은 S3C2440A 매뉴얼을 참조하면 알 수 있다.[1][4]

<표 1> LCD 컨트롤러 설정 항목

설정항목	설정값
LCD 패널 수직 사이즈	240-1
LCD 패널 수평 사이즈	320-1
디스플레이 모드	TFT LCD Panel
BPP 선택	16 bpp for TFT
16 bpp 출력 비디오 포맷	5:6:5 포맷
CLKVAL	5
프레임 버퍼 시작 메모리 주소	0x30800000
가상 스크린 메모리 크기	320*240 (가상스크린:LCD패널=1:1)
LCD 제어 DE/Sync 모드	Sync 모드
출력 해상도	320*240

마지막 단계인 파워-온(power-on)에서는 SPI를 통해 파워-온 명령을 LCD 장치로 전송함으로써 LCD 장치의 구동준비를 완료한다. LP35에 내장된 LTV350QV-F04 LCD는 파워-온과 파워-오프 순서를 제공하며, 16 진수 명령어를 매뉴얼에서 제시하는 지연 시간과 순서를 지켜 SPI를 통해 전송해야 한다. 이로써 TFT LCD를 구동하기 위한 초기 설정이 완료 된다.

4. 그래픽스 라이브러리 함수

픽셀이란 디지털 화면을 표현하는 최소단위로서 이 픽셀들이 모여 이미지를 형성한다. 그래픽 LCD에 이미지, 도형 등을 출력하기 위해서는 픽셀을 제어 하는 함수가 필요하다. TFT_PutPixel 함수는 픽셀 좌표(x, y)와 16비트 컬러 값(5:6:5)을 인자 값으로 받아서 해당 픽셀을 출력하는 함수이다.

S3C2440A LCD 컨트롤러는 시스템 메모리의 일부를 프레임버퍼로 사용한다. 프레임버퍼는 화면에 나타날 영상 정보를 일시적으로 저장하는 기억 장치이다. 프레임버퍼와 LCD 패널의 좌표체계는 서로 다르다. 따라서 LCD 패널의 XY 직교좌표를 LCD 패널에 대응되는 프레임버퍼의 선형적인 주소의 좌표로 변환해 주고, 프레임버퍼 배열에 그 좌표들의 색상값을 저장하는 루틴이 필요하다. 이는 TFT_PutPixel 함수에 의해 수행된다. 프레임버퍼의 해당 픽셀 주소에 비디오 데이터를 기록하면 LCD 전용 DMA 컨트롤러는 프레임버퍼 메모리에 있는 비디오 데이터를 CPU의 중재 없이 비디오 데이터 포트로 전송하는데, 그 결과 LCD 패널에 픽셀이 출력 된다.

본 논문에서는 픽셀 색상 값으로 16 bpp(5:6:5)을 사용한다. 입력색상을 위해 24 bpp(R(8):G(8):B(8)) 색상 값이 사용되므로 입력받은 24 bpp(8:8:8 형식) 색상 값을 16 bpp(5:6:5 형식)의 색상 값으로 변환한 후 사용해야 한다. 색상 값의 변환을 위해 R은 상위 5비트, G는 상위 6비트, B는 상위 5비트만 추출하여 <표 2>처럼 5:6:5 형식을 만든다.

<표 2> 16 bpp(5:6:5 형식) 색상

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R7	R6	R5	R4	R3	G7	G6	G5	G4	G3	G2	B7	B6	B5	B4	B3

점, 선, 삼각형, 사각형은 그림의 가장 간단한 도형 구성 요소이며, 이들을 확장하여 여러 가지의 도형들을 출력할 수 있다. 선, 원, 삼각형, 사각형과 같은 도형과 비트맵 이미지 파일을 출력하기 위해서는 다음과 같은 함수들이 필요하다.

- 도형 출력 함수
  a. TFT_DrawLine() : 직선 출력 함수
  b. TFT_DrawTriangle() : 삼각형 출력함수
  c. TFT_DrawRect() : 사각형 출력 함수
  d. TFT_DrawCircle() : 원 출력 함수
- TFT_DrawBMP() : 비트맵 이미지 출력 함수

TFT_DrawLine(x0, y0, x1, y1, color) 함수는 선을 그리는 함수이며, 시작점과 끝점의 좌표(x, y)와 색상 값을 인자 값으로 입력 받는다. 본 논문은 선을 그리기 위하여 Bresenham 선 알고리즘[6][7]을 사용 한다. 방정식을 이용하여 선을 그리는 경우, 실수 연산으로 인해 느린 연산 속도를 가져온다. 이러한 문제점을 해결하기 위해 고안된 Bresenham 선 알고리즘은 정수 계산만을 이용하여 실제 직선에 가까운 좌표를 찾아 그리는 더 정확하고 효율적인 선 생성 알고리즘이다.

기울기의 범위가 0<m<1이고 x1>x0, y1>y0 인 경우, 판단 매개변수 변수 Pk는 시작점 X0부터 X축 방향으로 증가함에 따라 Y축 좌표를 결정하는 변수이며, Pk의 부호에 따라 해당 X축 좌표에 대응하는 Y축 좌표를 결정하게 된다. 이러한 처리는 끝점까지 이어지며, 실제 선에 근사한 선이 생성된다. 1보다 큰 양의 기울기를 갖는 선에 대해서는 X와 Y좌표의 역할을 바꾸어 해결할 수 있으며, 음의 기울기를 갖는 선은 X, Y평면의 팔분면과 사분면 사이의 대칭성을 이용하여 생성할 수 있다. 마지막으로 수평선(△Y=0), 수직선(△X=0), 대각선(|△X|=|△Y|)인 경우는 X, Y 좌표의 증감소를 이용한다.

삼각형과 사각형을 그리는 함수는 각 꼭지점과 색상값을 인자값으로 입력받아 각 꼭지점을 TFT_DrawLine 함수를 이용하여 연결시킴으로서 쉽게 구현 할 수 있다.

TFT_DrawCircle(xCenter, yCenter, radius, color) 함수는 원을 그리는 함수이며, 원의 중심점 좌표(x, y)와 반지름 그리고 색상 값을 인자 값으로 입력 받는다. 정수 계산연산만을 이용하여 원을 그리기 위하여 중간점 알고리즘(midpoint algorism)[6][7]을 사용한다. 중간점 알고리즘은 단위 간격에서 샘플링하고, 각 단계에서 지정된 원 경로에 가장 근접한 픽셀 위치를 결정 한다. 판단 변수의 부호에 따라 해당 X축 좌표에 해당 하는 Y축 좌표를 결정한다. 중간점 원 알고리즘을 이용하여 1사분면의 45℃-90℃ 사이의 원주를 구하고, Y좌표와 X좌표의 역할을 바뀌어 0℃-45℃사이의 원주를 구한 후, X축 대칭과 Y축 대칭을 이용하여 4분면에 대한 원의 경로를 구할 수 있다.

TFT_DrawBMP(x, y, fp) 함수는 비트맵 이미지 파일을 출력 하는 함수이며, 이지미의 출력 시작 좌표(x, y)와 비트맵 파일의 파일 포인터 fp를 인자 값으로 입력 받는다 파일 포인터를 이용하여 비트맵 파일의 헤더 영역에서 수직, 수평 크기와 한 픽셀 당 비트수를 구한다. 일반적으로 비트맵 파일의 픽셀 당 비트수는 16바이트, 24바이트를 사용 하며, 출력 장치의 설정에 맞춰 변환 시켜 주어야 한다. 수평, 수직의 크기 정보를 이용하여 첫 픽셀부터 마지막 픽셀까지 24바이트 크기의 색상 값을 읽은 후 일차원 주소체계인 비트맵 파일의 데이터 주소를 이차원 주소 체계인 픽셀주소로 변환 시킨다. 변환 된 주소와 색상 값을 인자 값으로 TFT_PutPixel 함수를 호출해 해당 좌표의 픽셀을 출력 시킨다.

5. 결론 및 향후계획

본 논문은 시랩시스사의 LN2440SBC 임베디드 보드에 부착된 LP35의 TFT LCD 구동을 위한 초기화 방법과 픽셀 제어를 위한 픽셀 디스플레이 함수에 대해 설명하고, 이 함수를 이용하여 유저 인터페이스와 데이터의 가시화를 위한 그래픽스 라이브러 함수에 대해 설명했다.

향후에는 기존의 그래픽스 라이브러리 함수에 현재 입력장치로 각광받고 있는 터치스크린 기능을 추가하여, 비주얼적이고, 유동적인 유저 인터페이스를 구현해 보고자 한다.

참고문헌

[01] 홍건표, 하동호, "ARM920T S3C24101, S3C2440A를 이용한 개발자들을 위한 ARM 프로세서", OHM, 2006

[02] Samsung Electronics, "Product Information LTV350QV-F04", Samsung Electronics, 2005

[03] 오삼권, "uC/OS-II를 탑재한 S3C2440A 싱글 보드 컴퓨터를 위한 모니터링 시스템의 구현", 호서대학교 논문집, 제 26권, 2007

[04] Samsung Electronics, "S3C2440A 32-BIT CMOS MICROCONTROLLER USER'S MANUAL Revision 1", Samsung Electronics, 2004

[05] John Catsoulis, "Designing Embedded Hardware, 2nd ED의 역서", 한빛미디어, 2007

[06] 최윤철, 임순범, "컴퓨터 그래픽스 배움터(개정판)", 생능, 2006

[07] Donald Hearn, M.Pauline Baker, "COMPUTER GRAPHICS의 역서", 아진, 2002