

Sterowanie wyświetlaczem bez kontrolera grafiki

Nowoczesne urządzenia elektroniczne rzadko kiedy mogą obyć się bez wyświetlacza. Najczęściej wiąże się to z koniecznością użycia dodatkowego lub wbudowanego kontrolera graficznego, który nie tylko zwiększa koszty aplikacji, ale również komplikuje jej budowę. Nowoczesne mikrokontrolery pozwalają w wielu przypadkach na uniknięcie tego problemu. Jeśli projektowany graficzny interfejs użytkownika ma być prosty, możliwe jest stworzenie wirtualnego kontrolera, którego obciążenie jednostki obliczeniowej sięga jedynie ok. 5%.

W ogólności, kontroler wyświetlacza potrzebny jest po to, by móc odpowiednio często wysyłać kompletne ramki pikseli obrazu. W praktyce częstotliwość odświeżania ekranu wynosi najczęściej ok. 60 Hz, co oznacza, że ilość przesyłanych danych na sekundę jest dosyć duża. Może się wydawać, że będzie to silnie obciążać jednostkę obliczeniową, ale w mikrokontrolerach takich, jak Microchip PIC32, które obsługują DMA (*Direct Memory Access*), obciążenie procesora nie przekracza 5%. Bezpośredni dostęp do pamięci sprawia, że większość czasochłonnych operacji wykonywanych jest z pominięciem jednostki obliczeniowej.

Podstawowe pojęcia

Kontroler przesyła do wyświetlacza informacje o kolorach pikseli z częstotliwością odświeżania, wyrażoną w hercach. Ilość danych koniecznych do przesłania zależy przede wszystkim od rozdzielczości ekranu: np. 320×240 pikseli. Częstotliwość przesyłania pikseli zależy od ich liczby, a więc rozdzielczości oraz odświeżania. Aby móc je sprawnie przesyłać, należy je przygotować wcześniej i zapisać w pamięci – w praktyce w RAMie, w postaci tzw. ramki. Trzeba też uwzględnić paletę kolorów wyświetlacza, wyrażoną w bitach. Typowo, jeśli wyświetlacz prezentuje 16-bitową gamę barw, po 5 bitów poświęconych jest na kolor czerwony i niebieski, a 6 na zielony.

Sposób użycia DMA

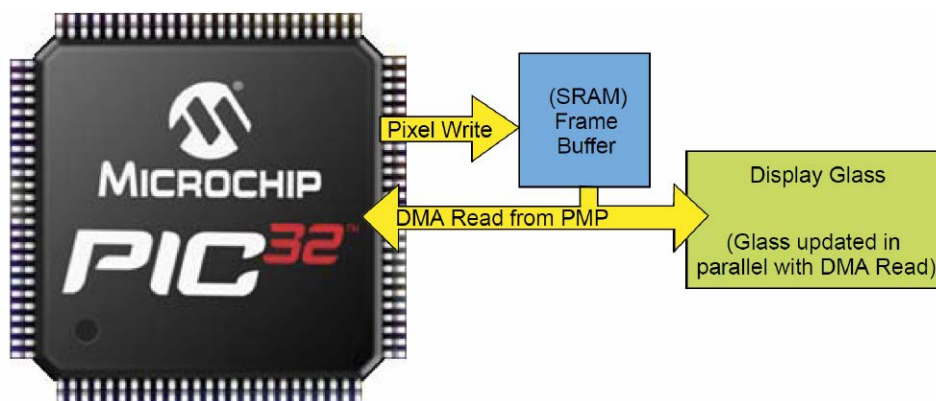
Obwody DMA pozwalają na przesyłanie danych z jednego miejsca do drugiego, bez ciągłej kontroli CPU. Wystarczy jedynie zaprogramować DMA tak, by przesyłało np. pojedynczą linijkę danych do wyświetlacza z użyciem nadrzędnego portu równoległego (PMP – *Parallel Master Port*). Po przesłaniu pojedynczej liniiki, PMP może wygenerować przerwanie, które spowoduje kolejny transfer DMA. Alternatywnie, w układach, które nie pozwalają na takie przerwanie, transfer DMA może być wyzwalany za pomocą timera.

Ponadto, w przypadku modeli z obsługą interfejsu zewnętrznej magistrali (EBI – *External Bus Interface*),

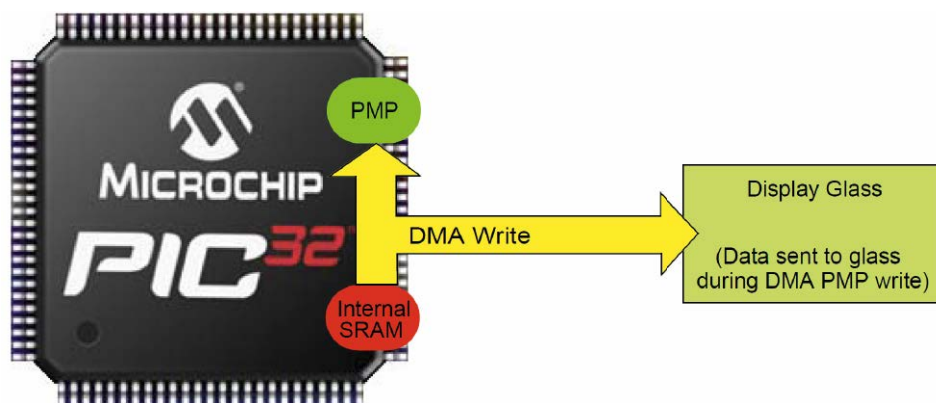
EBI może zostać wykorzystany jako zegar taktujący przesył danych pikseli. Zazwyczaj pozwala to uzyskać wyższą częstotliwość taktowania, niż przy wykorzystaniu portu PMP, przy czym EBI bazuje na tych samych wyprowadzeniach, co PMP.

Podczas transmisji danych, PMP lub EBI przesyła sygnał odczytu lub zapisu po każdym przesłanym, pojedynczym pikselu. W efekcie sygnały te służą jako zegar taktujący dla wyświetlacza. Po przesłaniu każdej linii danych, CPU otrzymuje przerwanie z DMA i konieczne jest zmodyfikowanie sygnałów sterujących synchronizacją pionową lub poziomą wyświetlacza (oznaczanych jako HSYNC i VSYNC) oraz sygnału DEN (*Data Enable*). Czynność ta jest powtarzana tyle razy, ile linii do wyświetlenia zawiera bufor ramki obrazu.

Bufor z ramką obrazu może znajdować się w pamięci RAM wewnętrznej lub zewnętrznej. Sposób komunikacji z użyciem zewnętrznej pamięci pokazano na **rysunku 1**. Wykorzystanie wbudowanej pamięci zilustrowano na **rysunku 2**.



Rysunek 1. Sterowanie wyświetlaczem z użyciem zewnętrznej pamięci RAM



Rysunek 2. Sterowanie wyświetlaczem z użyciem wbudowanej pamięci RAM

```

Listing 1. Sposób konfiguracji mikrokontrolera do pracy jako wirtualny kontroler wyświetlacza
//Zawieś wszystkie transfery DMA
DMACONbits.SUSPEND =1;
#define PMP_CONTROL
(PMP_ON|PMP_MUX_OFF|PMP_READ_WRITE_EN|\PMP_CS2_EN|\PMP_CS2_POL_LO|PMP_WRITE_POL_LO|PCLK_POLARITY
)
#define PMP_MODE
(PMP_DATA_LENGTH|PMP_MODE_MASTER2|\PMP_WAIT_BEG_1|PMP_WAIT_MID_1|PMP_WAIT_END_1)
// Przygotuj PMP
// PMP jest skonfigurowane do korzystania z linii danych i adresowych w trybie 16-bitowym
// Czasy oczekiwania PMP został skonfigurowane na najkrótsze
mPMPOpen(PMP_CONTROL, PMP_MODE, PMP_ADDRESS_LINES, PMP_INT_ON);
//Adres PMP ustawiany jest na 0
PMADDR = 0x0000;
// Otwarcie pożądanego kanału DMA
DmaChnOpen(1, 0, DMA_OPEN_DEFAULT);
// Określenie, jakie zdarzenie ma powodować rozpoczęcie transferu DMA
DmaChnSetEventControl(1, DMA_EV_START_IRQ(TIMER_2_IRQ));
// Określenie parametrów transferu: adresu źródłowego, adresu docelowego
// oraz rozmiarów; dla 16-bitowej palety rozmiar źródła to 2 bajty
DmaChnSetTxfer(1, (void*)&PMDIN, &GraphicsFrame[0], 2, HBackPorch, 2);
// Ustawienie priorytetu przerwania na 7 (najwyższy)
INTSetVectorPriority(INT_VECTOR_DMA(1), INT_PRIORITY_LEVEL_7);
// Ustawienie podrzędnego priorytetu przerwania
INTSetVectorSubPriority(INT_VECTOR_DMA(1), INT_SUB_PRIORITY_LEVEL_3);
// Włączenie przerwania wywoływanego w momencie zakończenia transferu
DmaChnSetEvEnableFlags(1, DMA_EV_BLOCK_DONE);
// Włączenie obsługi przerw w kontrolerze
INTEnable(INT_SOURCE_DMA(1), INT_ENABLED);
// Włączenie DMA
DmaChnEnable(1);
// Uruchomienie timera drugiego tak, by wyzwał zegar pikseli
OpenTimer2(T2_ON | T2_SOURCE_INT | T2_PS_1_1, 10);
// Rozpoczęcie wszystkich transferów DMA
DMACONbits.SUSPEND = 0;

```

Wyświetlacze TFT LCD

Omawiana metoda pozwala na sterowanie zarówno wyświetlaczami TFT LCD, jak i CSTN lub MSTN, o ile zostanie poddana drobnym modyfikacjom. Linie danych odpowiadają bitom potrzebnym na przesyłanie informacji o kolorach pikseli. Większość wyświetlaczy spotykanych na rynku ma od 8 do 24 linii danych, zależnie od palety barw.

Linie HSYNC, VSYNC, DEN i PCLK służą do synchronizacji przesyłanych danych pomiędzy ramką obrazu, a panelem wyświetlacza. Informują LCD o tym, kiedy zaczyna się i kończy linia (sygnał HSYNC), lub cała ramka (sygnał VSYNC). Sygnał DEN informuje wyświetlacz o tym, że dane są do niego przesyłane. Jest to wymagane przez niektóre z paneli TFT LCD, by te zdążyły przygotować się na przyjęcie informacji o pikselu.

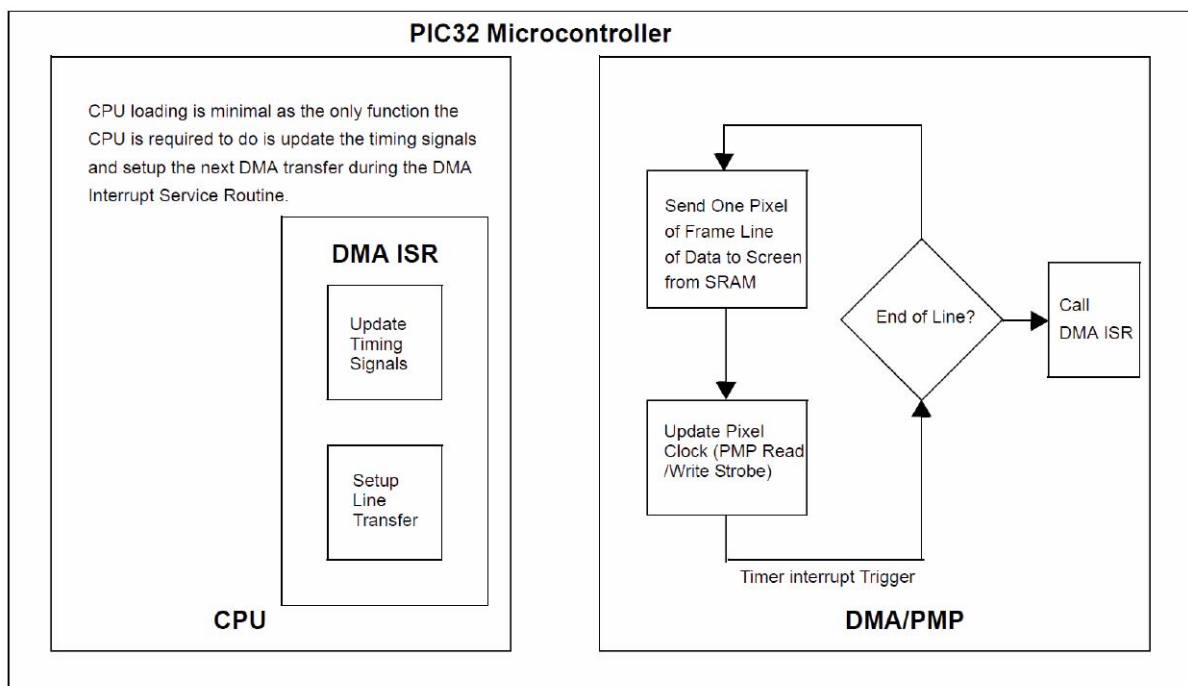
Każdy cykl zegara PCLK aktualizuje zawartość wyświetlacza. Pozostałe sygnały muszą być zsynchronizowane z sygnałem PCLK. W przypadku wyświetlaczy, które nie obsługują sygnałów

HSYNC ani VSYNC, wystarczy odpowiednie taktowanie z użyciem PCLK.

Przykład oparty o zestaw

Firma Microchip opracowała płytę Low-Cost Controllerless Graphics PICtail Plus daughter board (*LCC graphics board*), która pozwala zademonstrować opisaną technikę w praktyce i działa z wieloma istniejącymi zestawami do mikrokontrolerów PIC32. Przygotowane oprogramowanie ułatwia synchronizację i wymaga jedynie podania parametrów podłączanego wyświetlacza i obsługiwanych przez niego sygnałów. Wartości te trzeba wpisać do kodu i można go skompilować, by zacząć wyświetlanie ramek obrazu na ekranie.

Na **ilustracji 3** pokazano, co dzieje się wewnątrz mikrokontrolera, kiedy ramka obrazu jest przesyłana do wyświetlenia. Ilość zdań dla CPU jest znikoma, w porównaniu do obciążenia DMA. W praktyce konieczne staje się tylko napisanie kodu obsługiwanego przerwania



Rysunek 3. Operacje wykonywane przez CPU i DMA/PMP w trakcie pracy jako wirtualny kontroler grafiki

DMA oraz konfiguracji DMA i PMP. Przykładowy kod inicjalizacji został przedstawiony na **listingu 1**.

Przygotowywanie nowych pikseli

Wyświetlając obraz nie należy zapominać o konieczności aktualizacji samego bufora pamięci obrazu. To tam zapisywana jest treść do wyświetlenia i procesor musi zadbać, by była aktualna. Jeśli ramka zapisywana jest w pamięci zewnętrznej, transfery DMA są zawieszane na czas aktualizacji treści ramki. Jest to konieczne, gdyż mikrokontroler zawiera tylko jeden podukład PMP i jest on dzielony z wirtualnym kontrolerem grafiki. Oznacza to, że częstotliwość odświeżania wyświetlacza jest zaburzana przez aktualizację treści do wyświetlenia.

Z tego względu ważne jest, by liczba pikseli do aktualizacji była kontrolowana i minimalizowana. Pozwala to zapobiec niepotrzebnemu nadpisywaniu zawartości ramki takimi samymi danymi, które mogłyby na tyle opóźnić prezentację treści na wyświetlaczu,

że wszelkiego rodzaju animacje i zmiany przestałyby być płynne. Mechanizm kontroli można zrealizować poprzez wprowadzenie zmiennej zliczającej piksele do aktualizacji, która jest zerowana po każdym przerwaniu DMA.

Jeśli wykorzystane zostaną obwody EBI, czas potrzebny na aktualizację treści ramki jest silnie skrócony. Użycie EBI do zapisywania pozwala ograniczyć ilość przeladowywanych danych, gdyż rejestry PMP przestają być wykorzystywane zarówno do zapisu, jak i odczytu. Ponadto EBI po prostu pozwala na szybszą transmisję z zewnętrzną pamięcią RAM.

Oprogramowanie

Metoda opierająca się na wbudowanej pamięci SRAM korzysta z sygnału zapisu PMP jako zegara pikseli. W przypadku użycia płytki LCC graphics board konieczne jest włożenie zworek 1 i 2. Tryb ten w przygotowanej aplikacji korzysta z 8-bitowej palety barw. Natomiast zawartość wbudowanej pamięci SRAM jest stale zapisywana do PMP.

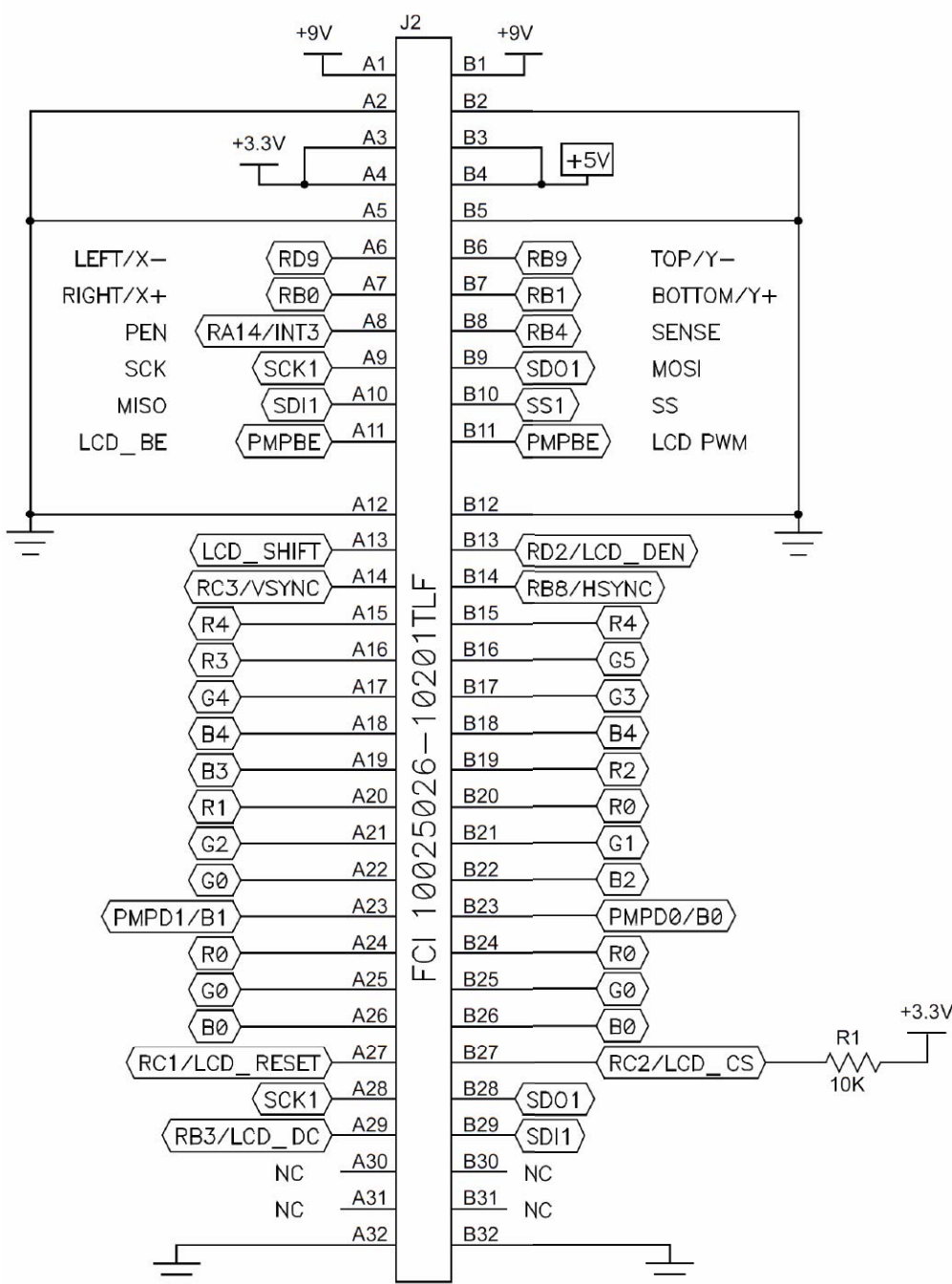
8-bitowa paleta w przypadku kolorowego obrazu oznacza w praktyce, że po trzy bity przeznaczone są na składową czerwoną i zieloną, a dwa pozostałe bity na kolor niebieski. Ten, jak i poprzedni sposób podziału bitów są powszechnie stosowane w wyświetlaczach ze względu na specyfikę ludzkiego oka, które jest najbardziej wrażliwe na kolor zielony, a następnie czerwony, a kolor niebieski rozróżnia najgorzej.

W przypadku metody z użyciem zewnętrznej pamięci, sygnał odczytu PMP jest sygnałem zegarowym pikseli. W tym celu należy wstawić zworki 2 i 3 na płytce LCC graphics board. W trybie tym stosowana jest 16-bitowa paleta barw, z podziałem składowym takim, jak podano wcześniej. Jeśli podłączany jest panel o więcej niż 16 liniach danych kolorów, nieużywane linie należy podłączyć do najbardziej znaczących bitów dla poszczególnych barw składowych. Dzięki temu możliwe będzie użycie pełnej skali barw – od czerni do bieli.

Podsumowanie

Jeśli prezentowany na ekranie obraz nie ma zmieniać się zbyt często, a rozdzielczość wyświetlacza nie jest bardzo duża, nawet zwykły, 32-bitowy mikrokontroler z łatwością samodzielnie poradzi sobie z obsługą ekranu, bez potrzeby dodawania dodatkowych podzespołów. Omówiona metoda pozwala więc uprościć projekt i zmniejszyć jego łączny koszt produkcji.

Marcin Karbowniczek, EP



Rysunek 4. Schemat połączeń płytki LCC graphic board z wyświetlaczem