

ISIX w przykładach w języku C

Obsługa klawiatury matrycowej

W ostatnim przykładzie wprowadzającym w programowanie ISIX-a, w języku C pokazemy w jaki sposób obsłużyć klawiaturę matrycową. Klawiatury tego typu dzięki multipleksowaniu pozwalają na obsługę wielu klawiszy przy stosunkowo niewielu wymaganych liniach I/O. W klasycznych rozwiązaniach bez systemu mikroprocesorowego, obsługa klawiatury matrycowej najczęściej jest realizowana przez procedurę obsługi układu czasowo-licznikowego.

W przypadku systemu operacyjnego, procedurę obsługi klawiatury numerycznej możemy delegować do specjalnie w tym celu utworzonego wątku. W bieżącym przykładzie do linii GPIO dołączymy telefoniczną klawiaturę matrycową 4×4 oraz znany z wcześniejszych przykładów wyświetlacz z telefonu N3310. Wątek ob-

sługi klawiatury w momencie wykrycia wciśnięcia klawisza będzie wysyłał wiadomość zawierającą kod klawisza, która będzie odczytywana i wyświetlana na ekranie N3310 przez wątek wyświetlania. Do realizacji powyższego przykładu konieczne będzie dołączenia modułu KAmo-
modLCD1 oraz modułu klawiatury matry-

Listing 1. Funkcja main() programu obsługi klawiatury matrycowej

```
int main(void)
{
    //Create ISIX blinking task
    isix_task_create( blinking_task, NULL, ISIX_PORT_SCHED_MIN_STACK_DEPTH, TASK_
    Prio_LED);
    //Create fifo msgs
    fifo_t *key_fifo = isix_fifo_create( 10, sizeof(key_t) );
    if(key_fifo)
    {
        //Create isix tasks (key and disp)
        isix_task_create(kbd_task, key_fifo, TASK_STK_SIZE, TASK_Prio_KEY);
        isix_task_create(display_srv_task, key_fifo, TASK_STK_SIZE, TASK_Prio_DISP);
    }
    isix_start_scheduler();
    //Start the scheduler
}
```

List. 2. Wątek obsługi klawiatury

```
ISIX_TASK_FUNC(kbd_task, entry_params)
{
    //Initialize gpio port keyscan
    keyscan_init();
    //Fifo pointer
    fifo_t *key_fifo = (fifo_t*)entry_params;
    //Main loop
    for(;;)
    for(int row=KEYSCAN_FIRST_ROW; row<=KEYSCAN_LAST_ROW ; row<=&=1)
    {
        //Set row output
        io_set_clr_mask( KEYSCAN_PORT, row, KEYSCAN_ROW_PINS );
        //Wait one tick before read
        isix_wait(1);
        int col = io_get_mask( KEYSCAN_PORT, KEYSCAN_COL_PINS);
        if(col>0)
        {
            //If col selected translate key and sent to disp
            key_t key = key_translate( row, col );
            isix_fifo_write("key_fifo", &key, isix_ms2tick(KEY_SCAN_INTERVAL));
        }
        //Wait one ms
        isix_wait_ms(KEY_SCAN_INTERVAL);
    }
}
```



cowej KAmo-
modLCD1 oraz modułu klawiatury matry-

Opis działania programu

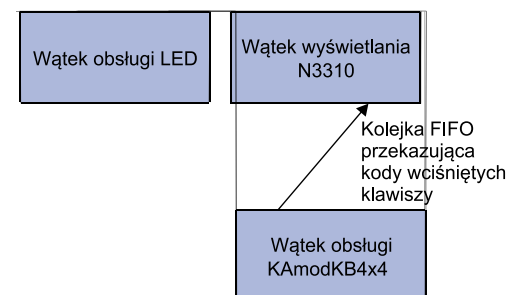
Zasadę działania programu z uwzględnieniem podziału na wątki przedstawiono na **rysunku 1**.

Wykonywanie aplikacji rozpoczyna się od funkcji main() - **listing 1**. Na początku tworzony jest niezależny wątek obsługi LED D1, a następnie kolejka FIFO składająca się z 10 kodów klawiszy. Jeżeli proces tworzenia kolejki został wykonany pomyślnie, to jest tworzony wątek obsługi klawiatury oraz wątek wyświetlania. Następnie jest uruchamiany planista zadań systemu ISIX. Za obsługę klawiatury odpowiada wątek *kbd_task* (**listing 2**).

Na początku wykonywana jest procedura inicjalizująca klawiaturę (**listing 3**), która wykonuje konfigurowanie linii kolumn PE0..PE3 w kierunku wejścia, natomiast linii wierszy PE4..PE7 w kierunku wyjścia.

Funkcja konfigurowania portów jest realizowana przez procedurę *io_config_ext*, która umożliwia skonfigurowanie grupy pinów w wybranym porcie, przekazanych przez maskę bitową.

Po wykonaniu czynności początkowych wątek przechodzi do pętli nieskończonej, gdzie następuje cykliczna zmiana stanów wyjść ROW, a następnie odczyt bitów linii COL. Na podstawie znajomości aktualnie aktywowanego stanu na liniach



Rysunek 1.

Listing 3. Inicjalizowanie linii I/O do obsługi klawiatury

```
//Initialize the port configuration
static void keyscan_init(void)
{
    RCC->APB2ENR |= RCC_APB2Periph_GPIOE;
    //Input push pull +GND
    io_config_ext( KEYSKAN_PORT, KEYSKAN_COL_PINS, GPIO_MODE_INPUT, GPIO_CNF_IN_PULLUP );
    io_clr_mask(KEYSCAN_PORT, KEYSKAN_COL_PINS);
    //Output row to zero
    io_config_ext( KEYSKAN_PORT, KEYSKAN_ROW_PINS, GPIO_MODE_10MHZ, GPIO_CNF_GPIO_PP );
    io_clr_mask(KEYSCAN_PORT, KEYSKAN_ROW_PINS);
}
```

Listing 4. Funkcja zwracająca kod wciśniętego klawisza

```
static inline char key_translate(int row, int col)
{
    switch(row) /* Select row */
    {
        case KEYSKAN_ROW1_BIT:
            switch(col) /* Select col */
            {
                case KEYSKAN_COL1_BIT: return '1';
                case KEYSKAN_COL2_BIT: return '2';
                case KEYSKAN_COL3_BIT: return '3';
                case KEYSKAN_COL4_BIT: return 'A';
            }
            break;
        case KEYSKAN_ROW2_BIT:
            switch(col) /* Select col */
            {
                case KEYSKAN_COL1_BIT: return '4';
                case KEYSKAN_COL2_BIT: return '5';
                case KEYSKAN_COL3_BIT: return '6';
                case KEYSKAN_COL4_BIT: return 'B';
            }
            break;
        case KEYSKAN_ROW3_BIT:
            switch(col) /* Select col */
            {
                case KEYSKAN_COL1_BIT: return '7';
                case KEYSKAN_COL2_BIT: return '8';
                case KEYSKAN_COL3_BIT: return '9';
                case KEYSKAN_COL4_BIT: return 'C';
            }
            break;
        case KEYSKAN_ROW4_BIT:
            switch(col) /* Select col */
            {
                case KEYSKAN_COL1_BIT: return '*';
                case KEYSKAN_COL2_BIT: return '0';
                case KEYSKAN_COL3_BIT: return '#';
                case KEYSKAN_COL4_BIT: return 'D';
            }
            break;
    }
    return 'x'; /*Unknown key
}
```

ROW (wierszy), oraz badania stanu linii wejściowych kolumn COL (kolumn), możemy ustalić kod wciśniętego klawisza. Po ustawieniu linii wierszy odczyt linii kolumn następuje z opóźnieniem jednego cyklu systemu operacyjnego (1 ms), tak aby dać czas na ustalenie się sygnałów wyjściowych. W przypadku wykrycia stanu wysokiego na jakiegokolwiek linii COL wywoływana jest funkcja *key_translate()*, która na podstawie aktualnego wiersza

0-1 komunikatu powitalnego, a następnie przejścia do pętli głównej wątku. W pętli za pomocą funkcji *isix_fifo_read()*, odczytywana jest kolejka FIFO znaków. Jeżeli w kolejce znajduje się jakiś znak, kod odczytanego znaku wyświetlany jest na wyświetlaczu LCD. Jeżeli w kolejce nie ma żadnego znaku, wówczas wątek jest usypiany.

Lucjan Bryndza

Listing 5. Wątek odczytujący stan klawiszy

```
static ISIX_TASK_FUNC(display_srv_task, entry_params)
{
    fifo_t *temp_fifo = (fifo_t*)entry_params;
    key_t key;
    nlcd_init(); /*Initialize LCD
    //Put welcome string
    nlcd_put_string( "www.boff.pl", 0, 0 );
    nlcd_put_string( "Clicked key:", 0, 1 );
    for(;;)
    {
        //Read data from fifo
        if(isix_fifo_read( temp_fifo, &key, ISIX_TIME_INFINITE )==ISIX_EOK)
        {
            nlcd_set_position(6,2);
            nlcd_put_char(key);
        }
    }
}
```

i aktualnej kolumny zwraca kod wciśniętego klawisza (**listing 4**).

Działanie funkcji tłumaczącej jest bardzo proste i sprowadza się do zwrócenia odpowiedniego kodu klawisza na podstawie aktualnego wiersza oraz aktualnej kolumny wykorzystując switch-case.

Tak przekształcony kod klawisza jest następnie wysyłany do kolejki FIFO, po czym jest wywoływana funkcja *isix_wait_ms()*, która usypia wątek odczytu klawiatury na 9 ms. Za odczytywanie stanu klawiszy odpowiada wątek *display_srv_task* (**listing 5**). Działanie wątku rozpoczyna się od inicjalizacji biblioteki obsługi wyświetlacza NCD3310, wyświetlenia w liniach

UKŁADY INTERNETOWE

AVT966
Karta przełączników sterowana przez Internet



Dostępne wersje:
A - płytki drukowane i dokumentacja
B - komplet elementów z płytką
C - układ zmontowany i uruchomiony

AVT953
Karta wejść z interfejsem Ethernet



Dostępne wersje:
A - płytki drukowane i dokumentacja
B - komplet elementów z płytką
C - układ zmontowany i uruchomiony

AVT927
Uniwersalny interfejs Internetowy



Dostępne wersje:
A - płytki drukowane i dokumentacja
B - komplet elementów z płytką
C - układ zmontowany i uruchomiony

www.sklep.avt.pl

Producent: AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 11
tel. 022 257 84 50, fax 022 257 84 55
e-mail: handlowy@avt.pl