Features of software development for the PalmOS operating system

Belykh A.A., Belykh I.A., Pokrovsky K.S.
(MPEI (TU), Moscow, Russia)

The PalmOS operating system was developed in 1996 and was used for organizers or personal digital assistants (PDAs). It should be noted that this OS was designed for a very specific hardware part - the DragonBall processor family (DSP) from Motorola. And, perhaps, for this all this time, PalmOS has not undergone major changes (at present, the 5th version of the OS has been released, in which significant changes have only been made).

This paper discusses the use of a PDA to control diagnostic and therapeutic devices and display the measurement results. The control system and display of results by means of a PDA must have the following technical characteristics:

- display on the PDA screen of the electrical conductivity values obtained from the controlled diagnostic module (apparatus) at the measurement points, as well as the value of the “drop” of the arrow for diagnosing the patient's condition, with a frequency of at least 50 times per second in the form of an arrow and a graph;
- ensuring the receipt of incoming electrical conductivity values at a rate of at least 100 times per second and smoothing them according to the principle of a floating average over 10 samples;
- selection of the required frequencies, pulse intensity, therapy time and other parameters both in manual and automatic modes. In manual mode, the doctor himself can enter a fixed frequency, intensity, therapy time and other parameters. In automatic mode, the doctor can select ready-made sets of therapy parameters from those available in the database using the menu;

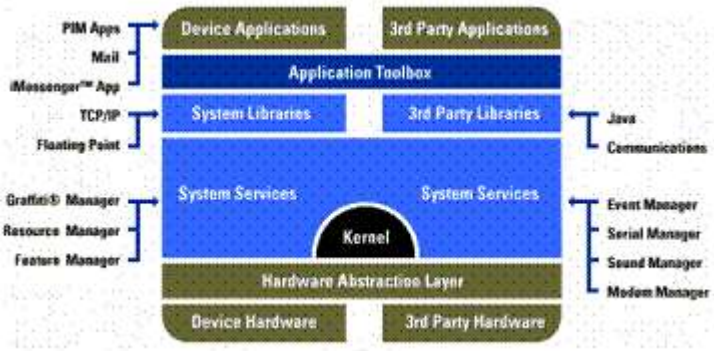
- the size of the database on therapeutic effects should contain up to 1000 therapeutic effects;
- the ability to connect to a PC for subsequent computer processing of diagnostic results;
- selection of the method of interaction between the PDA and a personal computer, both via wired and wireless interfaces.

The hardware part of the PDA. The DragonBall digital signal processor is built on a 16-32 MHz crystal with many built-in peripherals - serial I / O ports (UART), PWM (often used in devices for issuing simple audio signals), a touch screen driver, pins for controlling external memory, etc. The PDA itself can currently be equipped with a color (from 4 thousand to 65 thousand colors) or monochrome (from 4 to 16 shades of gray) touch screen with a resolution from 160x160 pixels to 320x320 pixels; a number of serial ports of input-output - COM (RS-232) and USB (at the moment - version 1.1, slave) through a cradle (on the PDA itself - the company's own connector, documented), infrared port; speaker (sound is output by the processor via PWM

module); built-in memory of data and programs from 1 MB to 16 MB; the presence of an expansion slot; 6-10 function keys and an area for handwriting information. Thus, we can make a preliminary conclusion that the data of the PDA (Palm) meet the stated requirements for the hardware.

The software part of the PDA (OS PalmOS). PalmOS is an OS with a microkernel (Micro Kernel OS), the main quality of this type of OS is a guaranteed short response time to any external event (AMX micro-kernel, developed by Kadak). The PalmOS operating system is a single-tasking operating system. At the same time, the formulated numerical requirements do not require a multitasking OS, and the nature of applications does not require multitasking. PalmOS applications are developed with one-thread, event-driven
oriented. Therefore, it would be natural to implement the reception of diagnostic data from the device (for processing and display) upon receipt of the entire data packet, that is, when the event of receiving the data packet is "triggered". However, the underdevelopment of the system for managing the data buffer from the serial port (the use of some functions for receiving data and a number of other functions in the event handler can lead to undesirable consequences - "freezing" of the PDA). Therefore, you have to periodically check (it is more logical to check when the application does not perform other actions) whether the next data packet has arrived, and only then send it for processing.

PalmOS consists of modules (Manager) and libraries (Library), for example, Memory Manager - memory management module, Data Manager - file system management module, TCP / IP library (see Fig. 1). A module is an integral part of PalmOS, and libraries can be added and removed.



Rice. one.Horizontal a slice of PalmOS and interaction with applications (from PalmOS SDK)

Note, what everyone module and library provides a documented API (Application Programming Interface) for applications. The operating system can automatically manage power.

An essential factor is the speed of displaying information on the display. It should not be less than the measurement data arrival rate. However, if you need to quickly display a large amount of graphic information (not tens, but hundreds and thousands of pixels), you need to use special methods (to directly speed up the output, and also to make this process less noticeable to the user).

The first way is to use an off-screen buffer. To create an off-screen buffer, you can use the function
WinCreateOffscreenWindow. After that, you need to save the current window on which drawing is performed using the function WinGetDrawWindow, and set the buffer in the form of a window on which drawing is performed, using the function WinSetDrawWindow. At the end of the work with the buffer, we restore the drawing window and, using the function
WinDrawBitmap (WinGetBitmap (offscreenBufferWinH), ...) mapping the offscreen buffer to the screen. This method gives good results when displaying a very large amount of graphical information. There is also a method related to direct access to the video buffer. Obtaining a pointer to a buffer is done as follows:BmpGetBits (WinGetBitmap (WinGetDisplayWindow ())). However, when writing directly to memory, on a PDA with a high resolution, if the application is designed only for 160x160, it will work only with the compatibility mode enabled, so when writing to the video memory you need
take into account that one pixel in pixel 160x160 on a 320x320 screen is represented by 4 mode.

Necessary Mark, what conclusion charts withusing of the proposed methods looks extremely bad on a PDA with a color display, especially with a resolution of 320x320 pixels. After all, you need to take into account that rendering in color is slower, therefore, when copying from the buffer to the screen, the interlaced strobe is clearly visible. In this case, instead of using a buffer, it is better to draw directly on the screen, having previously “locked” (fixed) it with the functionWinScreenLock (winLockDontCare), and after rendering "unlock" (unlock) WinScreenUnlock (). The display speed is the same, but the strobe is not visible.

The memory management system is one of the most important parts of PalmOS, which largely determines the architecture of the operating system and its features. Physically, the memory is located on memory cards (Cards), which are numbered - 0, 1,.... Each memory card can have RAM and ROM segments (RAM - random access memory, ROM - read only memory). PalmOS divides the entire RAM segment into Dynamic Heap and Storage Heap. Storage Heap is the equivalent of a disk (HDD), only files are there. Dynamic Heap is the equivalent of regular memory (Heap). The Dynamic Heap contains all the dynamic objects of the application, operating system, libraries, modules, as well as the stack (at the time of execution, the application code is not in the Dynamic Heap, i.e. all programs, like PalmOS itself, are always executed directly from the Storage Heap). The sizes of Dynamic Heap and Storage Heap are determined by the operating system during initialization and depend on the total memory size in the PDA. Typically, Dynamic Heap is 32–256 KB in size. You can work with Memory through the API of the Memory Manager module.

If the application requests a block of memory that is too large, and only small blocks are available, PalmOS will perform background defragmentation of the memory, and their base pointers will change due to the movement of existing blocks. So that the change of pointers does not affect the application, there are handles - pointers to a block pointer. A descriptor is the same pointer to memory, but in order to start working with this block

it is necessary to "block" it, i.e. get a pointer that will remain valid even if background memory defragmentation occurs. If the application allocates memory without using a descriptor
(MemPtrNew), either gets a pointer from a descriptor, i.e. "Blocks" it (MemHandleLock), then PalmOS will not move such a block during defragmentation. Despite the fact that the mechanism of descriptors is more complicated than just working with pointers, it cannot be avoided, because many of the system module API functions work with descriptors, i.e. care should be taken in advance that large data blocks in the program are represented in the form of descriptors. For a memory block, there is also such a thing as LocalID - a special number by which you can find out in which memory and on which memory card the memory block is located.

In addition, it should be noted that the processor performance in the mentioned PDAs is low - on average, one assembler instruction is executed in 4–6 clock cycles at 16 MHz (32 MHz in the latest devices). This may not be enough for resource-intensive preprocessing of measurement data.

In general, we can note a good combination of price and functionality of the device for the tasks of controlling diagnostic and therapeutic devices and displaying measurement results, the presence of a touch screen of the required size and various ways of pairing a PDA with devices, the ability to write programs in both high-level C / C ++ language and assembler.

Belykh A.A., Belykh I.A., Pokrovsky K.S. Features of software development for the PalmOS operating system // X

" IMEDIS ", 2004, vol. 2 - C.389-394