

УДК 811.111:621.382.049.77

A. Hmyrak, D. Korobeynik, Y. Beznis.

Microcontrollers. Programmable Logic Controllers

Belarusian National Technical University
Minsk, Belarus

Microprocessors and microcontrollers do not reveal any strict border between them as certain chips can access external code and/or data memory (microprocessor mode) and are equipped with particular peripheral components. Some microcontrollers have an internal RC (resistance-capacitance) oscillator and do not need an external component. However, an external quartz or ceramic resonator or RC network is frequently connected to the built-in, active element of the clock generator. Clock frequency varies from 32 up to 75 MHz. Another auxiliary circuit generates the reset signal for an appropriate period after a supply is turned on. Watchdog circuits generate chip reset when a periodic retriggering signal does not come in time due to a program problem [1].

There are several modes of consumption reduction activated by program instructions. Complexity and structure of the interrupt system (total number of sources and their priority level selection), settings of level/edge sensitivity of external sources and events in internal (i.e., peripheral) sources, and handling of simultaneous interrupt events appear as some of the most important criteria of microcontroller taxonomy. Although 16- and 32-bit microcontrollers are engaged in special, demanding applications (servo-unit control), most applications employ 8-bit chips. Some microcontrollers can internally operate with a 16-bit or even 32-bit data only in fixed-point range—microcontrollers are not provided with floating point unit (FPU).

New microcontroller families are built on RISC (Reduced Instruction Set) core executing due to pipelining one instruction per few clock cycles or even per each cycle. One can find further differences in addressing modes, number of direct accessible registers, and type of code memory (ranging from 1 to 128 KB) that are important from the view of firmware development.

Flash memory enables quick and even in-system programming (ISP) using 3–5 wires, whereas classical EPROM makes chips more expensive due to windowed ceramic packaging. Some microcontrollers have built-in boot and debug capability to load code from a PC into the flash memory using UART (Universal Asynchronous Receiver/Transmitter) and RS-232C serial line. OTP (One Time Programmable) EPROM or ROM appear effective for large production series. Data EEPROM (from 64 B to 4 KB) for calibration constants, parameter tables, status storage, and passwords that can be written by firmware stand beside the standard SRAM (from 32 B to 4 KB).

The range of peripheral components is very wide. Every chip has bidirectional I/O (input/output) pins associated in 8-bit ports, but they often have an alternate function. Certain chips can set an input decision level and pull-up or pull-down current sources. Output drivers vary in open collector or tri-state circuitry and maximal currents. At least one 8-bit timer/counter (usually provided with a prescaler) counts either external events or internal clocks, to measure time intervals, and periodically generates an interrupt or variable baud rate for serial communication. General purpose 16-bit counters and appropriate registers form either capture units to store the time of input transients or compare units that generate output transients as a stepper motor drive status or PWM (pulse width modulation) signal. A real-time counter (RTC) represents a special kind of counter that runs even in sleep mode. One or

two asynchronous and optionally synchronous serial interfaces (UART/USART) communicate with a master computer while other serial interfaces like SPI, CAN, and I²C control other specific chips employed in the device or system [1].

Almost every microcontroller family has members that are provided with an A/D converter and a multiplexer of single-ended inputs. Input range is usually unipolar and equal to supply voltage or rarely to the on-chip voltage reference. The conversion time is given by the successive approximation principle of ADC, and the effective number of bits (ENOB) usually does not reach the nominal resolution 8, 10, or 12 bits. There are other special interface circuits, such as field programmable gate array (FPGA), that can be configured as an arbitrary digital circuit. Microcontroller firmware is usually programmed in an assembly language or in C language. Many software tools, including chip simulators, are available on websites of chip manufacturers or third-party companies free of charge. A professional integrated development environment and debugging hardware (in-circuit emulator) is more expensive. However, smart use of an inexpensive ROM simulator in a microprocessor system or a step-by-step development cycle using an ISP programmer of flash microcontroller can develop fairly complex applications.

A programmable logic controller (PLC) is a microprocessor-based control unit designed for an industrial installation (housing, terminals, ambient resistance, fault tolerance) in a power switchboard to control machinery or an industrial process.

A PLC consists of a CPU with memories and an I/O interface housed either in a compact box or in modules plugged in a frame and connected with proprietary buses. The compact box starts with about 16 I/O interfaces, while the module design can have thousands of I/O interfaces. Isolated inputs usually recognize industrial logic, 24 V DC or main AC

voltage, while outputs are provided either with isolated solid state switches (24 V for solenoid valves and contactors) or with relays. Screw terminal boards represent connection facilities, which are preferred in PLCs to wire them to the controlled systems. I/O logical levels can be indicated with LEDs near to terminals. Since PLCs are typically utilized to replace relays, they execute Boolean (bit, logical) operations and timer/counter functions (a finite state automaton). Analog I/O, integer or even floating point arithmetic, PWM outputs, and RTC are implemented in up-to-date PLCs. A PLC works by continually scanning a program, such as machine code, that is interpreted by an embedded microprocessor (CPU). The scan time is the time it takes to check the input status, to execute all branches (all individual rungs of a ladder diagram) of the program using internal (state) bit variables if any, and to update the output status. The scan time is dependent on the complexity of the program (milliseconds or tens of msec). The next scan operation either follows the previous one immediately (free running) or starts periodically.

Programming languages for PLCs are described in IEC-1131-3 nomenclature: LD (Ladder Diagram), FBD (Function Block Diagram), ST (structured Text), IL (Instruction List), SFC (Sequential Function Chart). PLCs are programmed using cross-compiling and debugging tools running on a PC or with programming terminals (usually using IL), both connected with a serial link [1].

References:

1. Mode of access: [http://www.amci.com/tutorials/tutorials-what-is-programmable-logic-controller.asp/](http://www.amci.com/tutorials/tutorials-what-is-programmable-logic-controller.asp) Date of access: 10.02.2016