Method for Synchronizing a Group of Heterogeneous Microcontrollers with Time Control of Synchronous Work

Artem V. Osintsev, Alexander A. Sobko,
Maxim E. Komnatnov, *Member, IEEE*
Department of Television and Control
Tomsk State University of Control Systems and Radioelectronics, TUSUR
Tomsk, Russia

Maxim P. Sukhorukov, Yuriy A. Shinyakov
Scientific Research Institute of Space Technologies
Tomsk State University of Control Systems and Radioelectronics, TUSUR
Tomsk, Russia

***Abstract*— A method to synchronize a group of microcontrollers with different clock frequencies of their own generator is proposed. The algorithm of each microcontroller operation in the group and the block diagram of the group consisting of *N* microcontrollers are presented. Timing diagrams of signals of synchronous operation and clock pulses of the microcontrollers group are given. The result of the work, at the first stage, is the counting of received clock pulses by microcontroller software, which allows synchronizing the microcontrollers performing a task using the different system frequencies. In the second stage, the use of a precision real-time clock and of an «AND» gate ensures high synchronization accuracy at the time of the start and end of the work. A program that calculates the number of clock pulses for each microcontroller of the synchronized group based on the clock frequency of the microcontroller, on the frequency of the clock generator and on the period of synchronous operation is developed. The calculated coefficients can be used in the development of the microcontroller synchronization code. Using hardware and software and two synchronization stages, it became possible to obtain high accuracy and flexibility in setting the synchronous operation of a group of heterogeneous microcontrollers.**

Keywords*— *synchronization, microcontroller, heterogeneous computer network

I. Introduction

Microcontrollers (μC) are widely used inside radioelectronic equipment. This was facilitated by a large variety of the model series of computing architectures (8–32 bit), characteristics of computing and peripheral devices, communication interfaces (I^2C, 1-Wire, SpaceWire, RS-232, RS-485, CAN, USB and etc.). The use of at least two μCs expands the functionality and increases the computing power of a radioelectronics equipment, and also allows to work with a large number of peripheral devices. An increase in the number of tasks assigned to a radioelectronic equipment, for example, when controlling a large number of spaced devices, designing data acquisition systems from sensors, etc., requires the synchronization of a group of μCs, each of which performs a particular task. Therefore, there is a need for a group of μCs to perform actions with a certain periodicity, in connection with which the problem of synchronizing their work arises. Synchronization is necessary when performing tasks in real time [1]. Well-known methods for synchronizing μCs involve the use of precision quartz oscillators. There are methods of time synchronization using non-precision CMOS generators [2]. The synchronization problem is quite extensive [3], because clock generators, used in distributed locations are not absolutely accurate and do not support real-time accuracy [4]. One of the known synchronization methods is the use of a common external clock generator (ECG) for each μC in a group. This method is actively used [5, 6], however, it has many disadvantages such as: μC group should be of the same type; the work of the μC group is possible only on one frequency; an increase in the number of μCs greater than four leads to unstable operation of the μC due to the clock waveform distortion due to the difficulty of maintaining the signal integrity. The deficiency of control of synchronous operation and means of regulation of the period of synchronous operation of the μCs group also causes limitations on the possibility of using the existing synchronization methods. Methods for synchronizing the work of the μCs group of different manufacturers, at different system frequencies, are not known to the authors. Thus, it is useful to perform an analysis of the software and hardware of the μC and to develop a method that eliminates the above disadvantages. The purpose of the work is to develop a software-hardware method for synchronizing the operation of two or more μCs with the ability to control the period of synchronous operation.

The work is done as part of the RF Government Resolution implementation from 09.04.2010 №218 and agreement between Reshetnev Company "ISS" and The Russian Ministry of Education and Science from 01.12.2015 № 02.G25.31.0182.

II. Principle of The Proposed Method

A method for synchronization of the μC group is proposed, which allows to agree on the performance of a common task by a group of μCs that are not connected by an interface. μCs of synchronized group are not divided into "master" and "slave" (Master-to-Slave). At the same time, each μC performs its own instructions (code), without affecting the work of other μC groups. The block diagram of the device that implements the method of synchronizing the μC group is shown in Fig. 1.

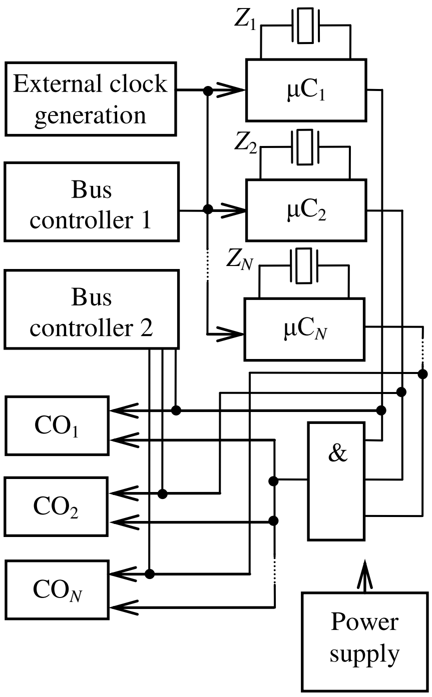

Fig. 1. The block diagram of the device synchronizing the μC group

The synchronization device allows to synchronize the operation of μCs from different manufacturers and perform flexible synchronization settings. Synchronization of μCs is carried out with minimal changes in the circuit of radio electronic devices. The result is achieved due to the use of each μC own (internal/external) clock, the counting each μC clock from an external clock, the setting the μC timers/counters, the confirmation signal (of the μCs group synchronization) through the AND gate.

The circuit (Fig. 1) operation principle is as follows. At the output of the external clock generator, the sheduled frequency (f_{ECG}) of clock pulses is set, which are applied to one of the digital inputs of the port of each μC in a group of N μCs. The μC input must support an external interrupt. The N-th μC operates at the desired $f_{\mu CN}$, frequency corresponding to the clock pulses generated by the internal or external clock from the internal or external (Z_N) crystal. The frequency f_{ECG} is given by the condition $f_{ECG} < f_{\mu CN}$, otherwise it will be impossible to calculate the number of clock pulses from an external clock (Fig. 2).

When an external interrupt is triggered by the μC, a function corresponding to this interrupt is called, which increments the counter variable in the memory of the μC. Thus, the received clock is counted from the external clock by each μC in the group.

As soon as the counter variable in the memory of the μC, responsible for the counting of the synchronous pulses (Fig. 3), reaches a certain value, an internal interrupt is triggered ($S_{\mu C1} - S_{\mu C4}$), responsible for the start of synchronous operation.

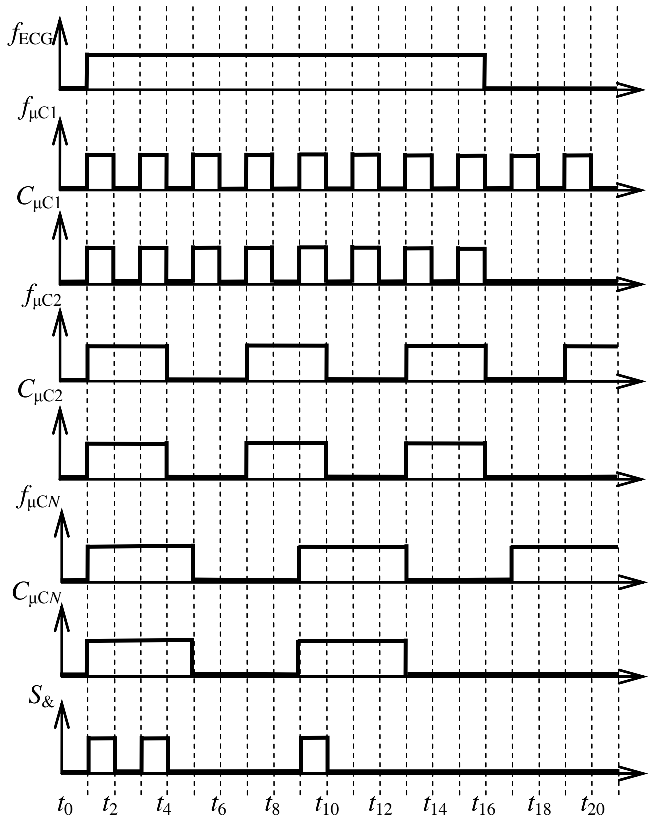

Fig. 2. Calculation of the number of sync pulses by a group of μCs

The S_{SCK} event at time t_3 (Fig. 3) means that all μCs are ready for synchronous operation. In this case, the following commands are successively executed: resetting the sync counter and starting the synchronous operation counter of the μC group with the time interval from t_3 to t_9 (Fig. 3). The synchronous operation counter can be implemented by the μC hardware timer, or programmatically in the μC's memory. The synchronous interval is calculated based on the $f_{\mu C}$, f_{ECG} and the specified period of synchronous operation.

At the time of t_9, when at least one μC from the group completes its operation, the synchronous operation ends (the desynchronization of a group of μCs). The following commands are executed: exit from the function of synchronous operation of a group of μCs; reset the synchronous period counter value; transition to the calculation of sync pulses. Fig. 3 shows, that the $S_{\mu C1}$ and $S_{\mu C3}$ completed the work at the same time, while the other μCs still continued to work. However, this result is no longer taken into account, since there is no signal from the AND gate.

All μCs can work on different frequencies and the synchronization count is calculated by everyone. Therefore, the onset of synchronous operation of each μC may differ by the period of time t_S between the readiness of the first and last μC in the group. It can be calculated as

$$t_S = |t_O - t_E| \quad (1)$$

where t_O – the start time of synchronous operation of $S_{\mu C(2N+1)}$; t_E – the start time for synchronous operation of the $S_{\mu C(2N)}$.

Reducing the time interval Δt to $t_N - t_{N-1}$ results in an improved synchronization of the μC group.

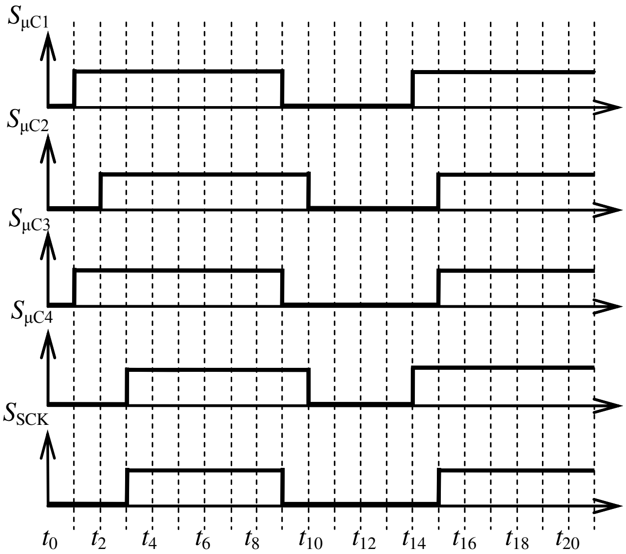

Fig. 3. Timing diagram of signals of synchronous operation of a gr. of μCs

III. Program Part

To calculate Δt, the software «Synchronizer for the operation of a group of μCs» was developed, which allows simulating the synchronous operation of a group of μCs (Fig. 4) [7]. The functions of the simulator are divided into two blocks. The first block calculates the number of sync pulses for each μC (Tabl. 1). In the second, a simulation of the work of a group of μCs occurs.

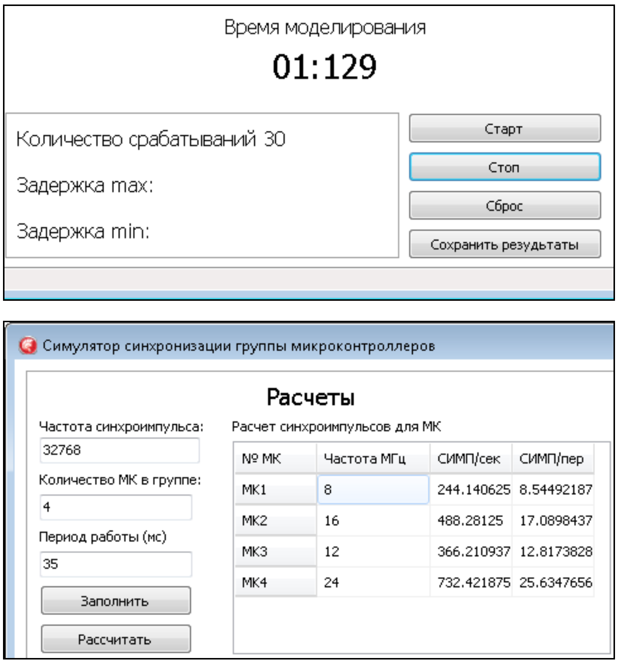

Fig. 4. Synchronization simulator (*a*) and a block for calculating the coefficients (*b*) for the synchronous operation of a group of μCs

Simulation allows to verify and optimize the algorithm without the need to develop the program code of the μC and shorten the time to develop a prototype device.

TABLE I. Calculations of The Simulator Program

μC	Frequency (MHz)	SIMP/s	SIMP/per
1	8	244.141	85.4492
2	12	366.211	128.174
3	16	488.281	170.898
4	24	732.422	256.348

As the initial data, the duration (ms) of the synchronous operation, the clock frequency and the number of μCs of the group are entered. Calculation of the optimum number of sync pulses (SIMP/s) for each μC is based on the frequency of the external clock and the μC. The calculated number of sync pulses for the specified synchronous operation period (SIMP/per) is used by the μC counter to determine the beginning of the common operation by calling the internal interrupt by raising the "start" flag in the μC code. Running the simulator allows you to check the calculated values. Thus, the duration of synchronous operation of a group of μCs is monitored, performing a common task. The algorithm for operation of each μC in a group is developed (Fig 5).

Based on the results of the simulation, for each μC the following data is recorded: the number of received clock signals; time of beginning and end of the general operation; the time interval between the first and last μC that started synchronous operation; the number of synchronous operations. Based on the data received in the simulator, a file with records of events with all information for the simulation period is formed. The received data can be saved on a computer to a file with the extension *.txt. To do this, the function «save results» is provided, which allows to analyze the results of the calculation in order to optimize the synchronizing the work of μCs group.

However, the simulator is limited in capabilities and can not perform the job with an accuracy comparable to the μC. The computer and μC commands differ in architecture and calculating capacity. It is impossible to provide all the factors of the code work in the μC. In addition to the synchronization algorithm code, there may be other tasks in the μC that affect the operation of the μC. Therefore, you can effectively manage the resources of the μC using real-time operating systems. In this case, the μC's tasks are managed by the scheduler (task manager), whose main task is to allocate equal time intervals and resources of the μC for the task in accordance with its priority.

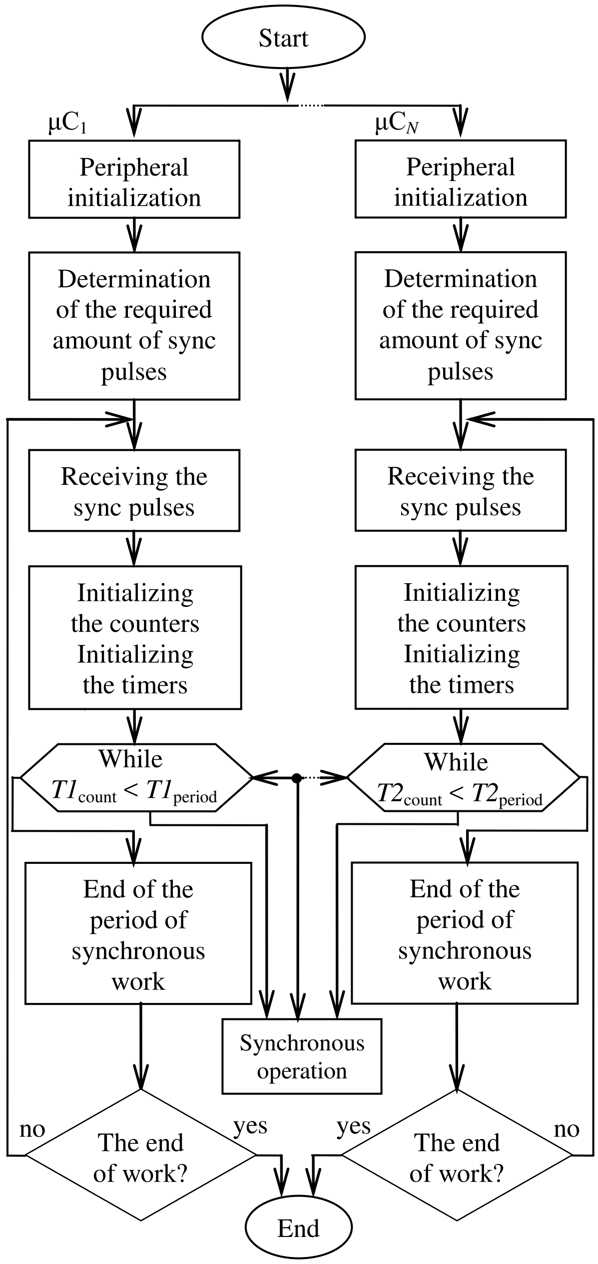

Fig. 5. The algorithm of operation of each μC in a group

IV. Conclusion

A hardware-software method for synchronizing the operation of a group of μCs is developed. It allows you to perform a task by a group of μCs, using minimal means. This method can be applied to ready-made devices in the form of a separate module, or by modifying the scheme of the final device. The versatility of the method lies in the fact that it allowes to synchronize the operation of μCs from different manufacturers. It is also possible to obtain a flexible synchronization setting by adjusting the duration of the synchronous operation period. The developed «Simulator of synchronous operation of a group of μCs» allows calculating the values of the timer coefficients of the counters of μCs and also check the calculated values by running the simulator, which allows to implement a certain mode (algorithm) for synchronizing a group of μCs.

References

[1] F. Terraneo, L. Rinaldi, M. Maggio , A.V. Papadopoulos, A. Leva "FLOPSYNC-2: efficient monotonic clock synchronisation," IEEE Symp. Real-Time Systems (RTSS 2014), Rome, Italy, pp. 1–4, December 2014.

[2] J. Boyle, J. Reeve, A. Weddell "DiStiNCT: Synchronizing Nodes with Imprecise Timers in Distributed Wireless Sensor Networks," IEEE Trans. on Industrial Informatics, V. PP, Iss. 99, pp. 1–9, March 2017.

[3] L. Lamport "Time, clocks, and the ordering of events in a distributed system," Comm. of ACM. vol.21, Iss. 7, pp. 558–565, July 1978.

[4] R.L. Probert, H. Yu, K. Saleh "Relative-clock-based specification and test result analysis of distributed systems," Conf. on Comp. and Communic., 1992. Conf. Proc., Eleventh Annual Int. Phoenix. Scottsdale, AZ, USA, pp. 687–694, April 1992.

[5] S.K. Phang, J.J. Ong, T.C.R. Yeo, B.M. Chen, T.H. Lee "Autonomous Mini-UAV for indoor flight with embedded on-board vision processing as navigation system," IEEE Region 8 Int. Conf. on Comput. Techn. in Electrical and Electronics Engineering (SIBIRCON), Irkutsk, Russia. – pp.722–727, July 2010.

[6] V.D. Yurkevich "Active Exoskeleton Tracking Control System," Int. Conf. on Biomedical Engineering and Comput. Techn. (SIBIRCON), Novosibirsk, Russia, pp. 124–129, October 2015.

[7] Certificate of state registration of the computer program №2017617018. Synchronizer for the operation of a group of μCs. Autors: Osintsev A.V., Gazizov T.R., Komnatnov M.E., Sobko A.A., Sukhorukov M.P. Application №2017613710. Recipt data 24 April 2017 г. Registered in the Register of Computer Programs 21 June 2017 г.