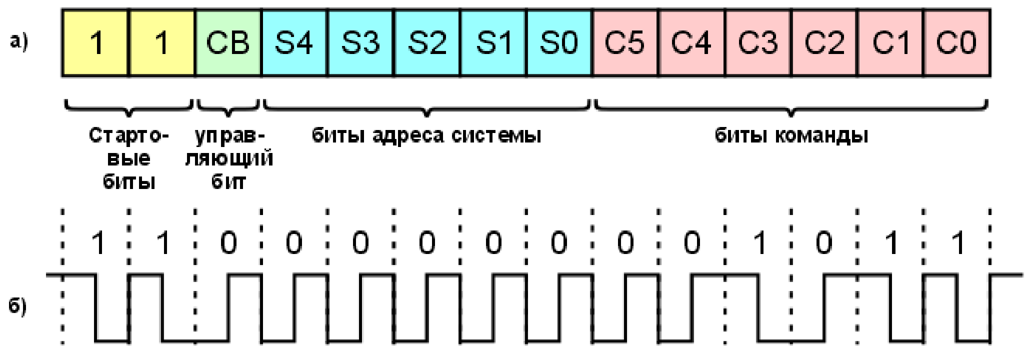
RC-5 code tester with error-correcting decoder

Remote control (RC) has long been a necessary attribute for various electronic equipment. The most commonly used remote control on infrared rays. One of the common ways to encode an IR message is the RC-5 code developed by Philips. The use of this code greatly simplifies the implementation of remote control, since remote controls operating in this standard, as well as transmitter microcircuits, are widespread. Almost any ready-made remote control can be adapted for your device by changing the RC-5 system number and making a new sticker with the necessary button labels. Integral IR remote control receivers are also quite common. All that remains is to decode the RC-5 code, which is usually done in software on the microcontroller. One of the decoder options is described below,

There are quite a few implementations of RC-5 decoders, one of them is even given in Application Note 410 from Atmel. However, software polling of the IR receiver output is most often used. Given the relatively long duration of the RC-5 burst, this results in a fairly large CPU load, which can interfere with other real-time tasks. The proposed decoder does not use software polling, but operates on interrupts. Interrupt handlers are very short, resulting in very little CPU usage. In addition, to improve noise immunity, the decoder implements multiple checks of the output status of the IR receiver within one transmitted interval, the true value is calculated according to the majority principle. The sampling moments are formed by a timer, the number of samples can be set to any odd number and is set at compile time. In practice, 3 samples are enough. If desired, you can work with the 1st sample by setting the appropriate value of the constant, this will further reduce the processor load. Additionally, the correctness of the "Manchester" code, which is used in the RC-5 standard, is checked. The decoder is written in C language and works on microcontrollers of the AVR family.



Rice. 1. The structure of the RC-5 package and the signal at the output of the photodetector.

The structure of the RC-5 package is shown in fig. 1a. The package includes 2 start bits, 1 control bit, 5 system number bits, and 6 command code bits. The start bits are designed to set the AGC in the IR receiver chip. The control bit is a sign of a new button press, it changes its state with each new press. The system number can take values 0…31, each value corresponds to its own equipment class. For example, 0 is a TV, 5 is a VCR, and so on. Many system numbers are not used by any standard equipment and can be used to control your equipment. Command code can take values 0…63, each remote control button has its own command code. Some command codes are standardized, making remote controls from different manufacturers compatible. When using the RC-5 code to control unique equipment, this standardization does not matter, it is only necessary to specify a system number that is not used by other equipment. In remotes built on the SAA3010 chip, this is done using a jumper (see below).

Later, to increase the number of commands, an additional command code bit (F bit) was introduced into the RC-5 standard, which takes the place of the second start bit in the package. This doubles the number of possible commands.

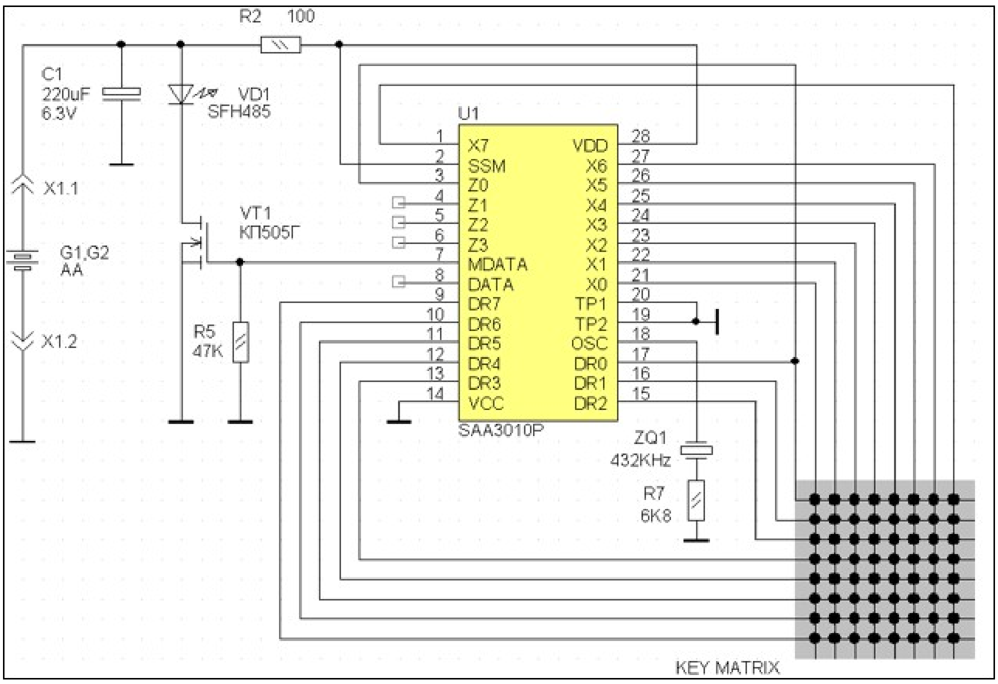
For data transmission in the RC-5 code, two-phase coding (Manchester code) is used. The duration of one cycle is 1.778 ms. As long as the button remains pressed, the message is transmitted at 64 clock intervals, i.e. 113.778 ms. The transmission uses 36 kHz carrier frequency modulation. Demodulation is performed by the IR receiver chip, the output is binary data without modulation. It must be borne in mind that the microcircuits of IR receivers incorporate a band-pass filter tuned to the carrier frequency. Therefore, only those microcircuits that have a filter center frequency of 36 kHz are suitable for receiving a RC-5 package. As a rule, IR receiver chips (SFH506, TFMS5360, TSOP1836 and others) have an inverted data output. An example of the signal at the output of the IR receiver microcircuit, taking into account the inversion, is shown in Fig. 1b.

To accurately measure time intervals on microcontrollers of the AVR family, a 16-bit timer capture circuit is usually used. In this case, this is not necessary, since the data transmission in the RC-5 package is at a low speed, moreover, during decoding, a deviation in the duration of intervals of up to 25% is allowed. Therefore, it is enough to use an external interrupt and an 8-bit timer, and leave the "deficient" 16-bit timer with a capture circuit free for other applications.

The decoder works in the following way: the signal from the output of the IR receiver is fed to the input of an external interrupt of the microcontroller. After the start bit is detected (a one-to-zero transition at the interrupt input), the timer 0 interrupt is enabled in the external interrupt handler and the interval up to the first T_SAMPLE sample is loaded. In the Timer 0 interrupt, samples are taken for each half bit. Samples are counted in the SampCnt variable. The number of samples is given by the SAMPLE_COUNT constant. The logic level for each half bit is calculated by the majority principle. To do this, the sum of the samples in the SampVal variable is calculated. If a HIGH level is detected at the input, then one is added to this variable, if LOW, it is subtracted. The sum value cannot be zero because the total number of samples is always set to odd. The first half of the current bit is used to decide on the value of the received bit. To check the correctness of the Manchester code, this level is compared with the value of the second half of the previous bit, which is stored in the variable PreVal. If the values match, then there was an error, and the reception starts from the beginning. The same happens if the next transition at the output of the IR receiver is not detected after the time T_SAMPLE * 2 after the last sample (timeout error). The received bits are pushed into the Rc5Code variable. The received bits are counted in the BitCounter variable. When the RC5_LENGTH bits are received, the reception is considered complete, the system number is copied to the SysVar variable, and the command code is copied to the ComVar variable. An additional command code bit (F bit) is also added to the ComVar variable. To check the correctness of the Manchester code, this level is compared with the value of the second half of the previous bit, which is stored in the variable PreVal. If the values match, then there was an error, and the reception starts from the beginning. The same happens if the next transition at the output of the IR receiver is not detected after the time T_SAMPLE * 2 after the last sample (timeout error). The received bits are pushed into the Rc5Code variable. The received bits are counted in the BitCounter variable. When the RC5_LENGTH bits are received, the reception is considered complete, the system number is copied to the SysVar variable, and the command code is copied to the ComVar variable. An additional command code bit (F bit) is also added to the ComVar variable. To check the correctness of the Manchester code, this level is compared with the value of the second half of the previous bit, which is stored in the variable PreVal. If the values match, then there was an error, and the reception starts from the beginning. The same happens if the next transition at the output of the IR receiver is not detected after the time T_SAMPLE * 2 after the last sample (timeout error). The received bits are pushed into the Rc5Code variable. The received bits are counted in the BitCounter variable. When the RC5_LENGTH bits are received, the reception is considered complete, the system number is copied to the SysVar variable, and the command code is copied to the ComVar variable. An additional command code bit (F bit) is also added to the ComVar variable. which is stored in the PreVal variable. If the values match, then there was an error, and the reception starts from the beginning. The same happens if the next transition at the output of the IR receiver is not detected after the time T_SAMPLE * 2 after the last sample (timeout error). The received bits are pushed into the Rc5Code variable. The received bits are counted in the BitCounter variable. When the RC5_LENGTH bits are received, the reception is considered complete, the system number is copied to the SysVar variable, and the command code is copied to the ComVar variable. An additional command code bit (F bit) is also added to the ComVar variable. which is stored in the PreVal variable. If the values match, then there was an error, and the reception starts from the beginning. The same happens if the next transition at the output of the IR receiver is not detected after the time T_SAMPLE * 2 after the last sample (timeout error). The received bits are pushed into the Rc5Code variable. The received bits are counted in the BitCounter variable. When the RC5_LENGTH bits are received, the reception is considered complete, the system number is copied to the SysVar variable, and the command code is copied to the ComVar variable. An additional command code bit (F bit) is also added to the ComVar variable. The received bits are pushed into the Rc5Code variable. The received bits are counted in the BitCounter variable. When the RC5_LENGTH bits are received, the reception is considered complete, the system number is copied to the SysVar variable, and the command code is copied to the ComVar variable. An additional command code bit (F bit) is also added to the ComVar variable. The received bits are pushed into the Rc5Code variable. The received bits are counted in the BitCounter variable. When the RC5_LENGTH bits are received, the reception is considered complete, the system number is copied to the SysVar variable, and the command code is copied to the ComVar variable. An additional command code bit (F bit) is also added to the ComVar variable.

The RC5_Init() function initializes the internal variables of the decoder and configures the hardware. This function must be called once before using the decoder.

The RC-5 transmitter can also be built on a microcontroller, but in most cases this is not necessary. Specialized RC-5 transmitter ICs are produced, the most common of which is the SAA3010. This chip is specially designed for building remote controls. It allows a supply voltage of 2 ... 7 V, the current consumption in standby mode does not exceed 10 μA. The chip is capable of scanning a matrix of 64 buttons and can transmit any of the 32 RC-5 code systems. Typically, the SAA3010 is used in single system mode by driving the SSM pin HIGH. In this mode, all commands are transmitted with a fixed system number. The system number is set by a jumper between pins Zi and DRj and is calculated using the following formula: SYS = 8i + j. command code, which will be transmitted when pressing the button that closes the Xi line with the DRj line, is calculated by the following formula: COM = 8i + j. An example of an IR remote control circuit based on the SAA3010 chip is shown in fig. 2.



Rice. 2. Diagram of the remote control based on the SAA3010 chip.

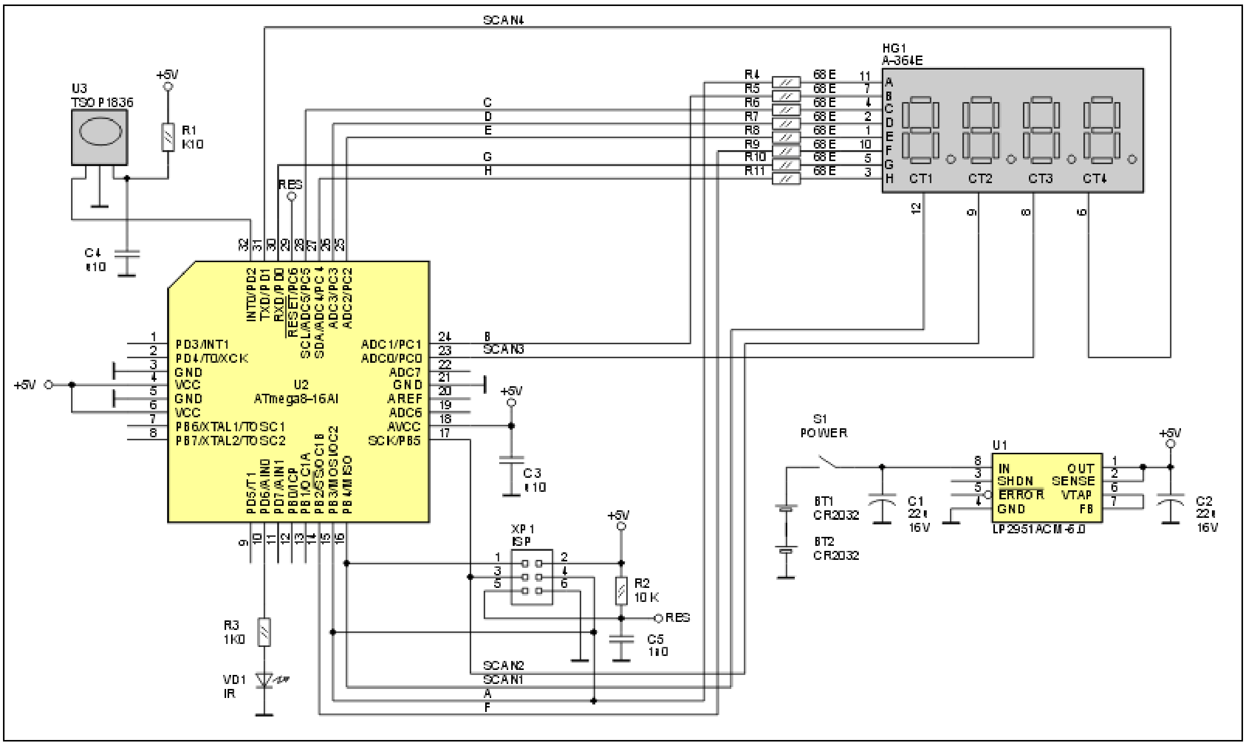
In the above diagram, system number 0 is set (used by the TV), for which a jumper is installed between pins Z0 and DR0. To change the system number, you need to remove this jumper and install a new one between the other pins Z and DR.

The OSC pin is the input/output of a 1-pin oscillator and is designed to connect a ceramic resonator at a frequency of 432 kHz. In series with the resonator, it is recommended to include a 6.8 kΩ resistor.

Test inputs TP1 and TP2 must be connected to ground during normal operation. A high logic level on TP1 increases the scan rate, and a high level on TP2 increases the frequency of the internal shift register.

At rest, the DATA and MDATA outputs are in the Z-state. The pulse train generated by the transmitter at the MDATA output has a duty cycle of 36 kHz (1/12 of the clock frequency) with a duty cycle of 25%. The DATA output generates the same sequence, but without padding. This output is used when the transmitter chip acts as a built-in keyboard controller. The signal at the DATA output is completely identical to the signal at the output of the remote control receiver chip (but, unlike the receiver, it has no inversion). Both of these signals can be processed by the same decoder. Using the SAA3010 as a built-in keyboard controller is in some cases very convenient, since only one interrupt input is consumed by the microcontroller to poll a matrix of up to 64 buttons.

As an example of using the described decoder, the following is a description of a stand-alone remote control tester based on the RC-5 code. The tester has an LED indicator that displays the system number (first and second digits), command code (second and third digits) and control bit value (dot between the second and third digits).



Rice. 3. Diagram of the RC-5 tester.

Принципиальная схема тестера приведена на рис. 3. Тестер построен на основе микроконтроллера ATmega8, который тактируется от встроенного RC-генератора частотой 8 МГц. ИК-приемник U3 типа TSOP1836 подключен к входу прерывания INT0. Питание приемника дополнительно фильтруется цепочкой R1C4. Через токоограничивающие резисторы R4-R11 к микроконтроллеру подключен светодиодный индикатор HG1, динамическая индикация реализована программно. Для индикации наличия ИК-посылки служит светодиод VD1. Он загорается даже в том случае, если посылка корректно не декодирована (например, формат посылки не совпадает с RC-5). Разъем XP1 служит для внутрисхемного программирования микроконтроллера. Питается тестер от двух литиевых элементов BT1 и BT2 типа CR2032 суммарным напряжением 6 В. С помощью LDO-стабилизатора U1 типа LP2951 получается напряжение питания 5 В. Нужно отметить, что возможно непосредственное питание устройства от одного элемента напряжением 3 В, но для этого понадобится установить низковольтную микросхему ИК-приемника, например, TSOP1836SS3V.

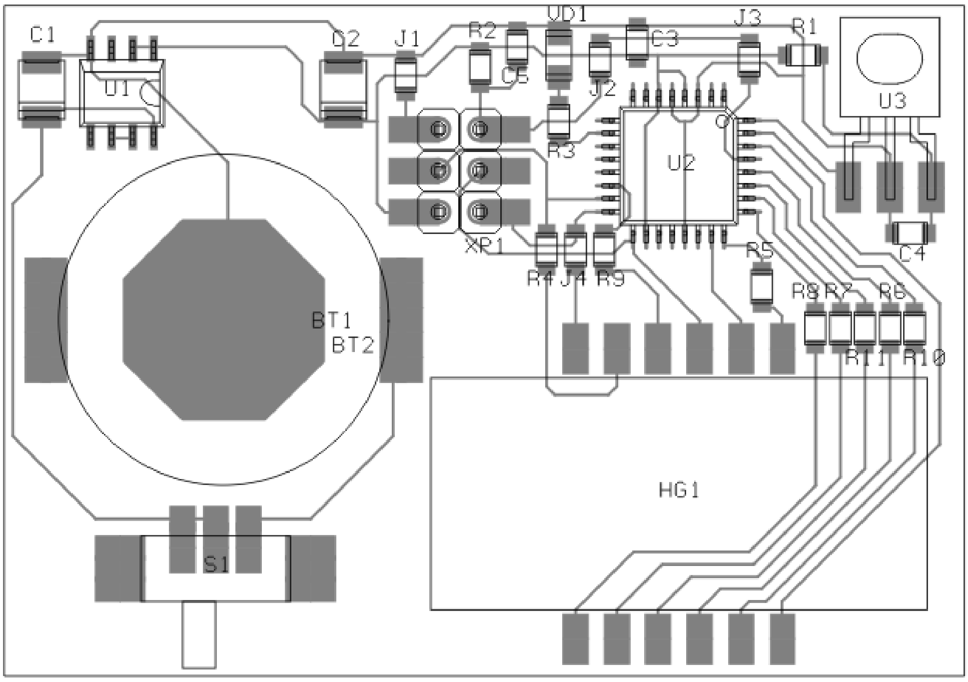


Рис. 4. Расположение элементов на печатной плате тестера RC-5.

На рис. 4 показано расположение элементов на печатной плате тестера. Односторонняя плата не имеет ни одного отверстия. При разводке использованы перемычки J1-J4 в виде SMD-резисторов размера 0805 нулевого номинала. Выводы фотоприемника, индикатора, разъема программирования и выключателя питания загнуты под прямым углом, чтобы обеспечить монтаж этих элементов на поверхности. Держатель элементов питания изготовлен из латунной пластинки шириной 5 мм. Внешний вид собранного тестера показан рис. 5.

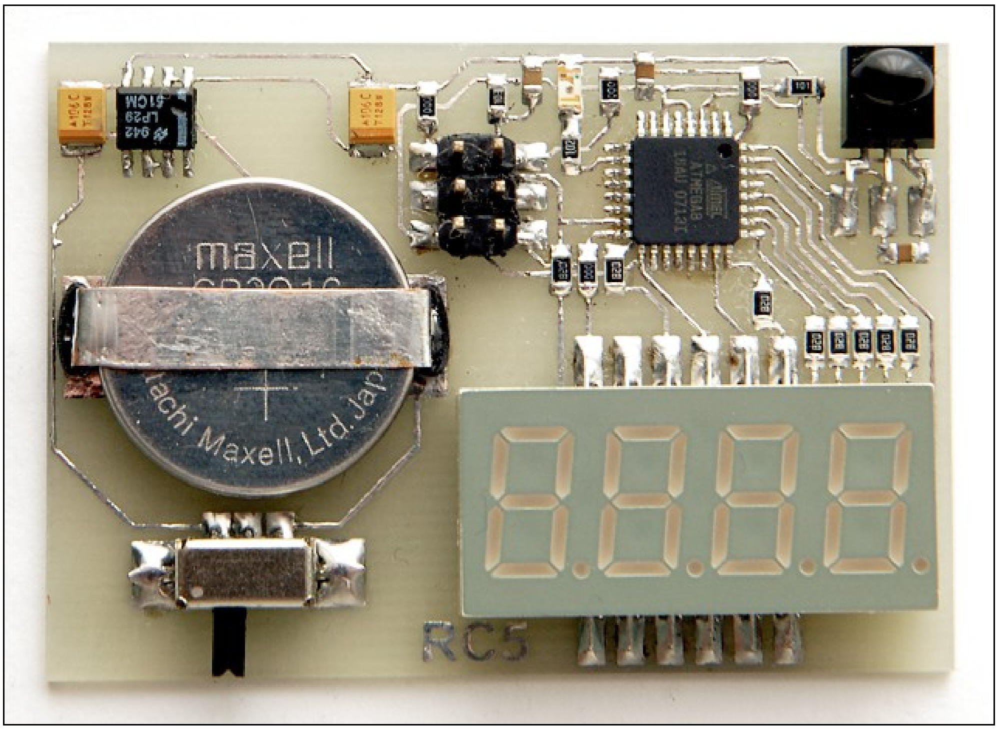


Рис. 5. Внешний вид тестера RC-5.

Downloads:

- Tester_sch.pdf (68 kB) - принципиальная схема тестера.
- Tester_pcb.pdf (49 kB) - рисунок печатной платы тестера.
- Tester_top.pdf (123 kB) - расположение элементов на плате тестера.
- Tester_source.zip (23 kB) - прошивка и исходник тестера (EWAVR).
- RC-5_tx.zip (21 kB) - пример передатчика RC-5 на AVR (AVR Studio).

Home > Projects > RC-5 Tester

Ridiko Leonid Ivanovich www.leoniv.diod.club e-mail: wubblick@yahoo.com