AVR415: RC5 IR Remote Control Transmitter on tinyAVR and megaAVR

APPLICATION NOTE

Introduction

Use of IR (infrared) light as a method for wireless communication has become popular for remote control applications. There are a number of different standards for such communication. In this application note the widely used RC5 coding scheme from Philips will be described, and a fully working remote control solution will be presented. This application will use the Atmel® ATtiny28 AVR® microcontroller for this purpose. This powerful unit contains a hardware modulator, a high current LED driver, and interrupt options, which makes it especially well suited for these kinds of applications.

Features

- Utilizes ATtiny28 special HW modulator and high current drive pin
- Size efficient code, leaves room for large user code
- Low power consumption through intensive use of sleep modes
- Cost effective through few external components
- Complementary of AVR410 RC5 IR Remote Control Receiver on Atmel tinyAVR® and megaAVR®

Table of Contents

1. RC5 Coding Scheme

The RC5 code is a 14-bit word bi-phase coded signal as seen in the figure below. The first two bits are start bits, always having the value “1”. The next bit is a control bit, which is toggled every time a button is pressed on the remote control transmitter. This gives an easy way of determining whether a button is pressed and held down, or pressed and released continuously.

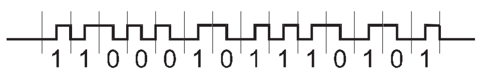
Figure 1-1. RC5 Frame Format

St_1	St_2	Ctrl	S_4	S_3	S_2	S_1	S_0	S_5	S_4	S_3	S_2	S_1	S_0

Five system bits hold the system address so that only the right system responds to the code. Usually, TV sets have the system address 0, VCRs the address 5, and so on. The command sequence is six bits long, allowing up to 64 different commands per address. The bits are transmitted in bi-phase code (also known as Manchester code) as shown below, along with an example where the command 0x35 is sent to system 5.

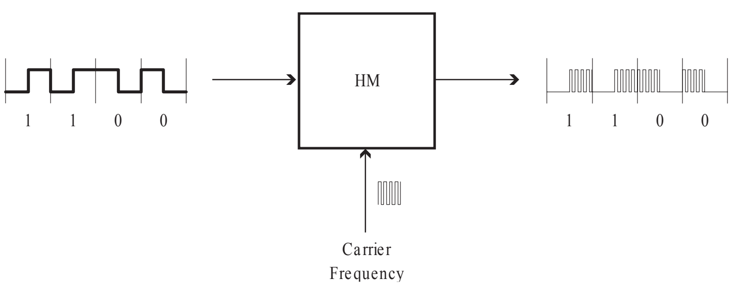
Figure 1-2. Bi-phase Coding

Figure 1-3. Example of Transmission



Note that the figures above show the signal that enters the Atmel ATtiny28 hardware modulator. The actual signal emitted by the IR-LED will be modulated with a certain carrier frequency as shown in the figure below.

Figure 1-4. Signal Before and After Modulation



2. ATtiny28: The Ideal Solution for Intelligent Remote Control Systems

ATtiny28 is a low-cost, high-performance 8-bit AVR RISC microcontroller with a number of features that makes it well suited for remote control applications. The built-in hardware modulator eases the task of generating the carrier frequency on which a data signal can be modulated. Frequency and duty-cycle are both easily changed by modifying the value residing in the Modulation Control Register MODCR. The high current driver on pin two of port A (PA2) is capable of driving an LED with a minimum of external components. This reduces size and system cost. In Power-down mode, the microcontroller can be configured to wake up on a low level from any pin on Port B. This provides an easy solution for waking up, scanning the keyboard, sending the command, and returning to Power-down mode. This application implements an easy keyboard scanning routine using Port B and Port D.

3. Implementation

The figure below shows the complete schematics for a remote control transmitter. See the table for component descriptions. The 455kHz resonator gives the application a reliable and flexible clock base. The external LED driver circuit provides a constant current for the IR-LED. Resistor R3 determines the driver strength, and is in this application chosen to 7Ω giving a drive capability of approximately 100mA. Higher resistor values will reduce current, and lowering the resistor value will increase driver strength. The diodes, D1 and D2, are present to ensure a close to constant driving current and to compensate for temperature variations in the transistor.

Figure 3-1. RC5 Transmitter

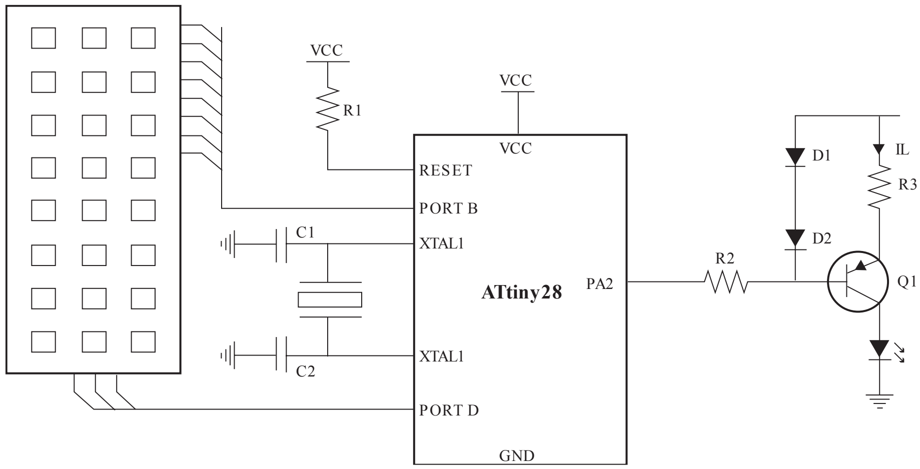


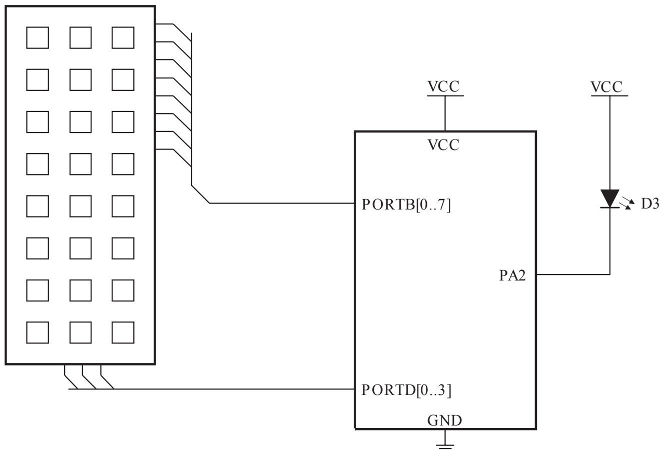
Table 3-1. Components

	Type	Comment
R1	3kΩ	External pull-up resistor is present to make the system less susceptible to external noise. Without this component, noise might reset the microcontroller.
R2	3kΩ	$R2 = \frac{V_{CC\min} - 1.4}{2I_L}\beta$
R3	7Ω	$R3 = \frac{0.7}{I_L}$
C1	100pF	Resonator dependent
C2	100pF	Resonator dependent
D1	1N4148	Small signal diode
D2	1N4148	Small signal diode
Q1	BC807-40TD	$I_{MAX} = 0.5A, \quad \beta = 250 - 600$
XTAL	455kHz	Resonator

4. Low Cost Implementation

For cost sensitive applications with high tolerance on accuracy, a solution utilizing the internally calibrated RC Oscillator of the ATtiny28 could be used. The high current drive capabilities of PA2 can sink the LED directly giving a solution with only a few external components as shown in the figure below.

Figure 4-1. Low Cost RC5 Transmitter



By using an external resonator and a driver circuit for the IR LED, a more flexible solution is achieved. The main advantage is higher driver capabilities and higher frequency stability over voltage range. If however the receiver is self-synchronized, it will adapt to the changing frequency of the transmitter, and a solution using the internal RC Oscillator could be used with good results.

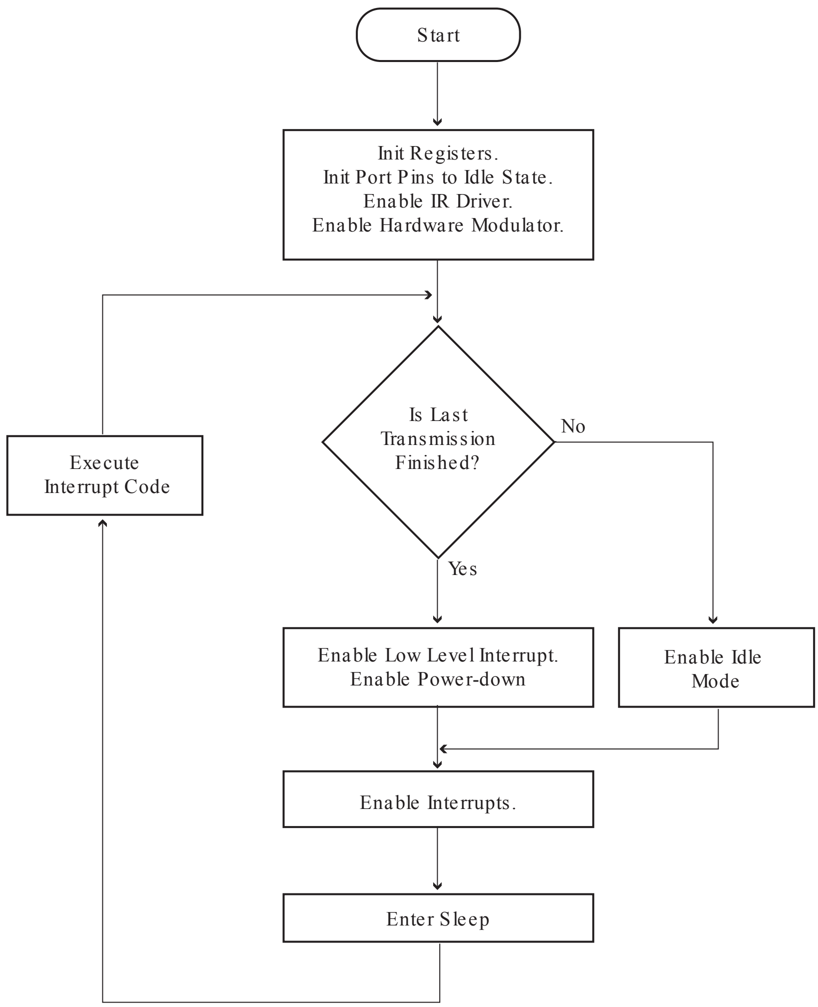
5. Software

The assembly code found in the AVR415.ASM file contains the latest RC5 Transmitter software. The program execution can be divided into two routines. Both of them are interrupt driven, and use different Powerdown modes to reduce power consumption. The program is designed to use only one level of hardware stack, leaving two levels for user code.

Main

The main program loop is shown in the figure below. First all registers are initiated. The hardware modulator is configured for correct frequency and duty-cycle. In this application 38kHz is used as the carrier frequency. This differs from the RC5 standard, which specifies 36kHz for the carrier wave. The RC5 signal will, however, be the same, and most standard RC5 Receivers should have no problem receiving and decoding the signal. Once the I/O modules are initialized the purpose of the main loop is to decide what sleep mode to use after the next wake-up.

Figure 5-1. Main Loop Flowchart

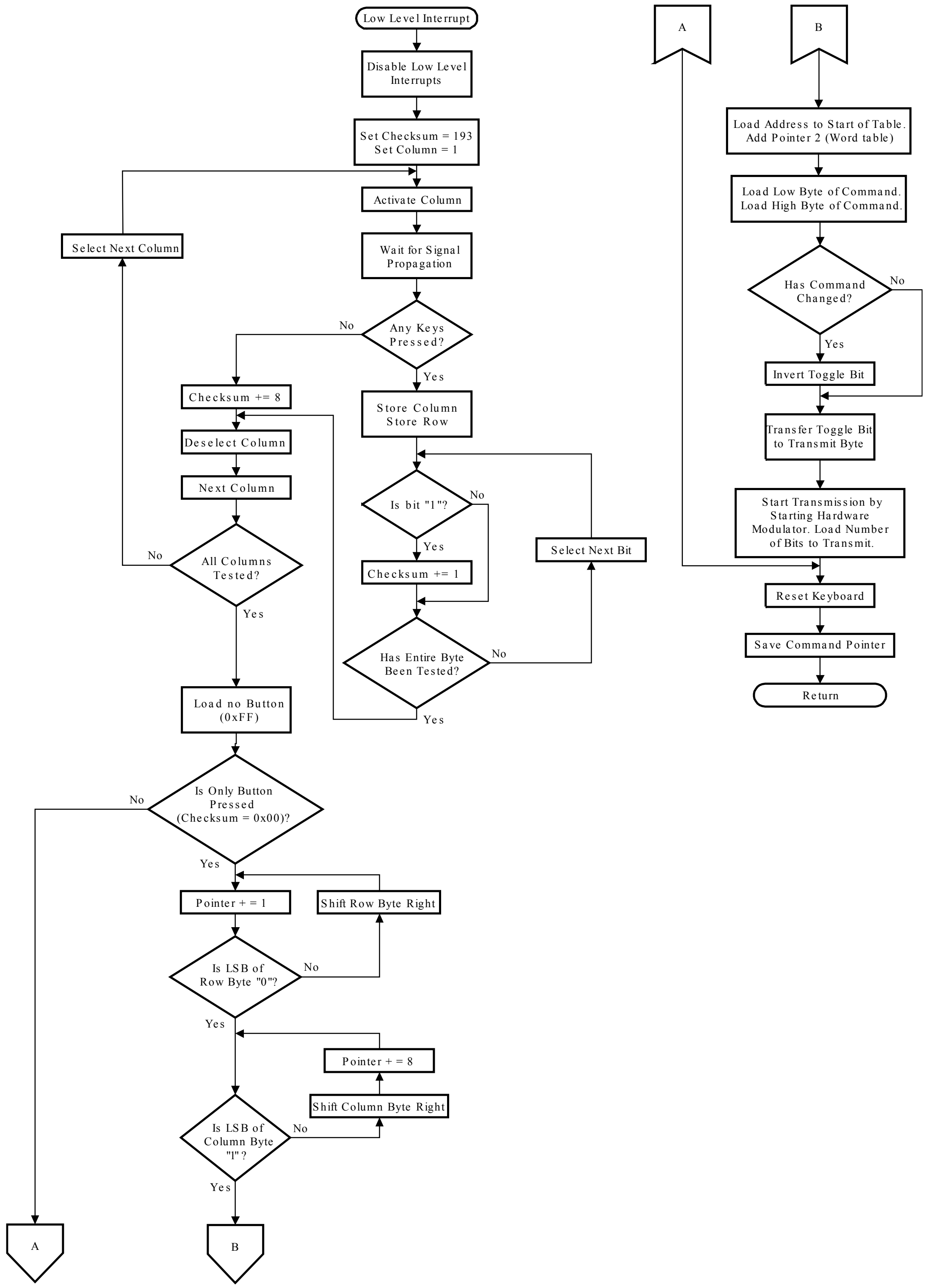


The program execution can roughly be divided into two states; “Transmitting a RC5 code” and “Waiting for a key to be pressed”. While waiting for a key to be pressed, the ATtiny28 is put in Power-down mode. In this mode the current consumption for the device is at a minimum, and the wake-up time is slightly longer than for the Idle mode. Since the wake-up condition is caused by physically pressing a key, the longer wake-up time will not cause a noticeable delay in the system.

Low Level Interrupt

When the ATtiny28 is in Power-down mode, a low level on any of the Port B pins will generate a Low Level interrupt, waking the device and executing the code illustrated by the flowchart in the figure below. The main purpose of this routine is to scan through the keyboard, and determine if a valid key is pressed. If two or more buttons are pressed simultaneously the routine returns the value 0xFF indicating an error. The “checksum” ensures that 63 of 64 combinations of row and column lines are high – that only one unique combination, representing the key, is low. If only one key is pressed, the column and row bit pattern is decoded into a pointer, which is used to perform a look-up in the Command table. Further, the Low Level interrupt also controls the toggling of the control bit, indicating if a new “instance” of a command is present, or if the “same” command should be re-transmitted. At the end of this routine, the hardware modulator is started preparing the transmission.

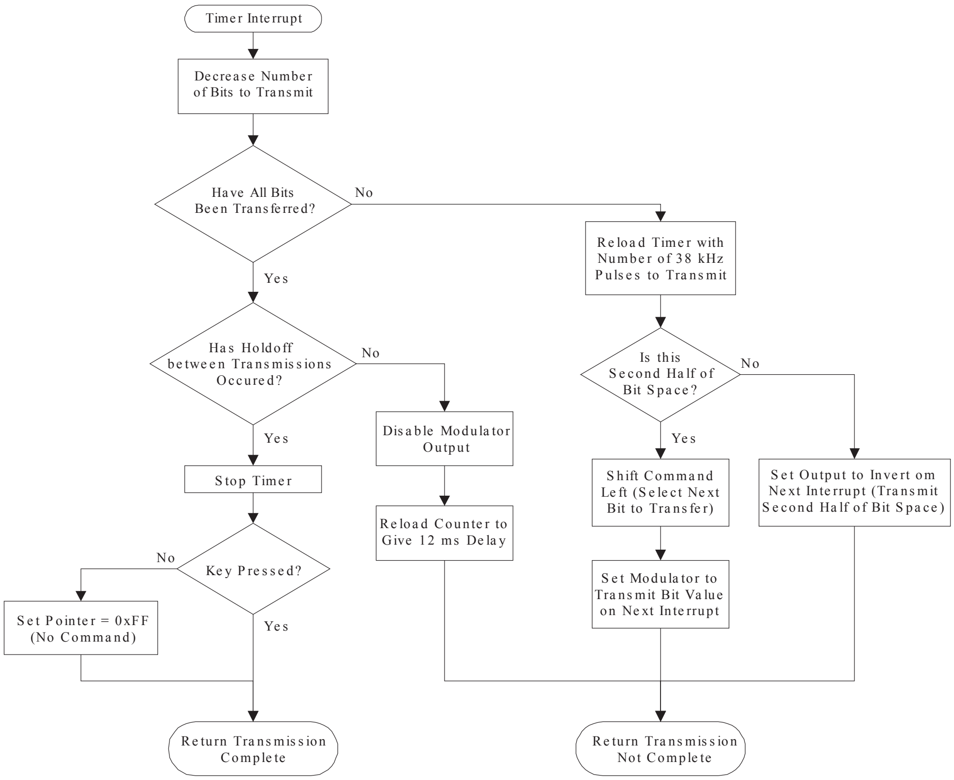
Figure 5-2. Low Level Interrupt Flowchart



Timer Interrupt Routine

The figure below shows the flowchart for the Timer Overflow interrupt. The main task of the Timer Interrupt routine is to keep track of the bit pattern that will be modulated on the IR LED, i.e., make sure that the transmitted signal is in accordance with the bi-phase coding scheme. Once a complete frame has been transmitted, the routines also generate a necessary delay before a new transmission is to be started.

Figure 5-3. Timer Overflow Interrupt Flowchart



6. Summary

This application note describes how to make a simple RC5 Transmitter. Due to the flexible hardware of the ATtiny28, other IR coding schemes could easily be implemented. It is also possible to change the duty cycle of the transmitted signal, further decreasing the power consumption and thus extending battery life. This application note acts as a foundation upon which other features can be implemented to result in a power efficient Remote Control Transmitter.

7. Revision History

Doc Rev.	Date	Comments
2534B	08/2016	New template and some minor updates
2534A	05/2003	Initial document release

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

© 2016 Atmel Corporation. / Rev.: Atmel-2534B-RC5-IR-Remote-Control-Transmitter-on-tinyAVR-and-megaAVR_Application Note-08/2016

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, AVR®, tinyAVR®, megaAVR®, and others are registered trademarks or trademarks of Atmel Corporation in U.S. and other countries. Other terms and product names may be trademarks of others.

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death ("Safety-Critical Applications") without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.