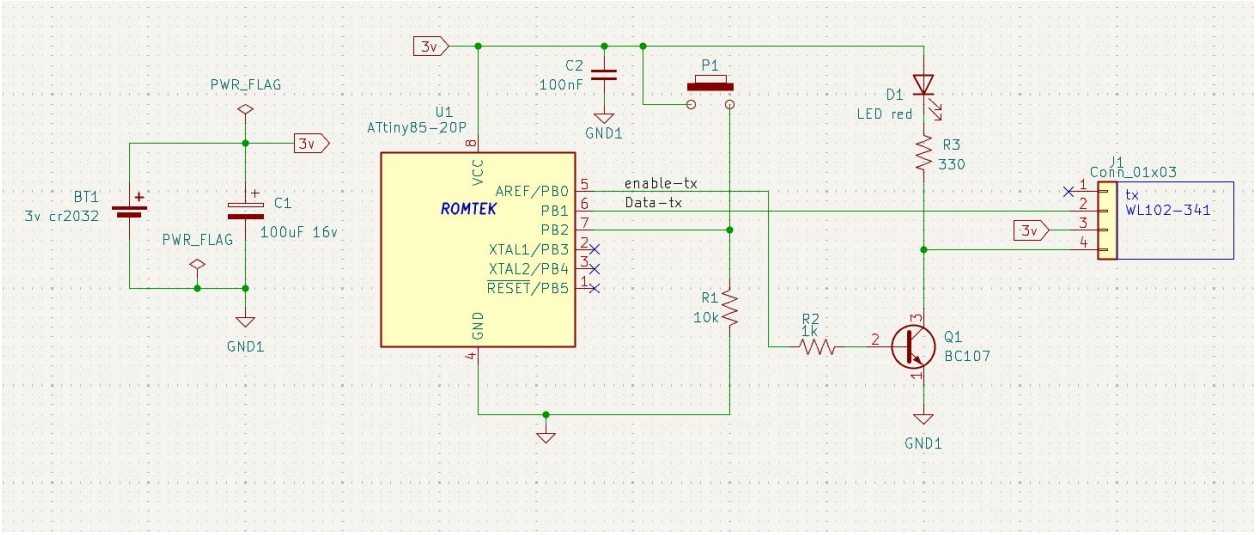
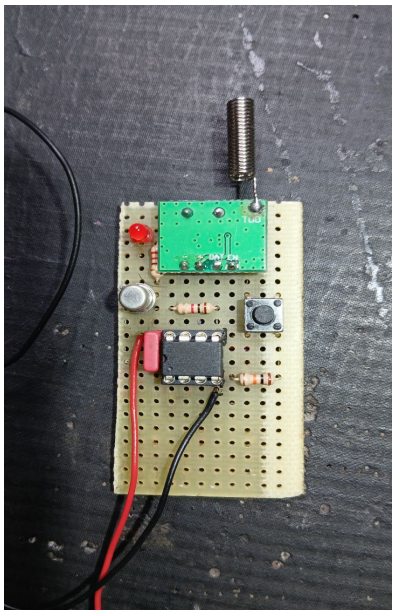
Example of an ask-ook transmitter and receiver

The purpose of this project is to demonstrate how to use ask-ook transmission using the low-cost WL101 and WL102 modules, with ATTINY85 microcontrollers.
The project is divided into two parts: the WL102 transmitter and the WL101 receiver.
The modules operate at a frequency of 433 MHz in ask-ook mode.

The transmitter is powered by a 3-volt battery.
Until the button is pressed, it is in very low power consumption. When the button is pressed, the microcontroller activates transistor Q1, which groundes the pin of the WL102 TX module, powering it and the LED. Then, again from pin 6 (Data-TX), it sends the string, then returns to standby.

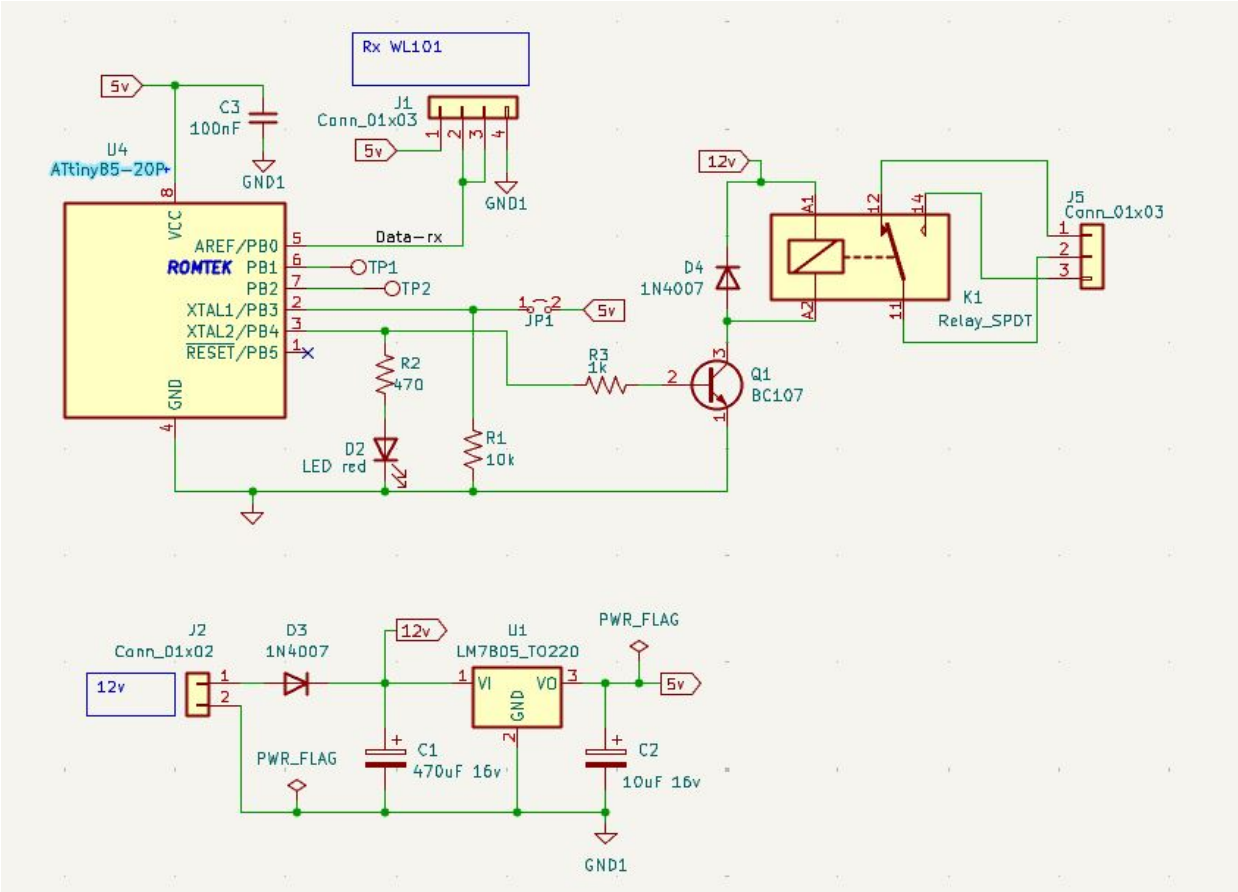
Transmitter diagram.

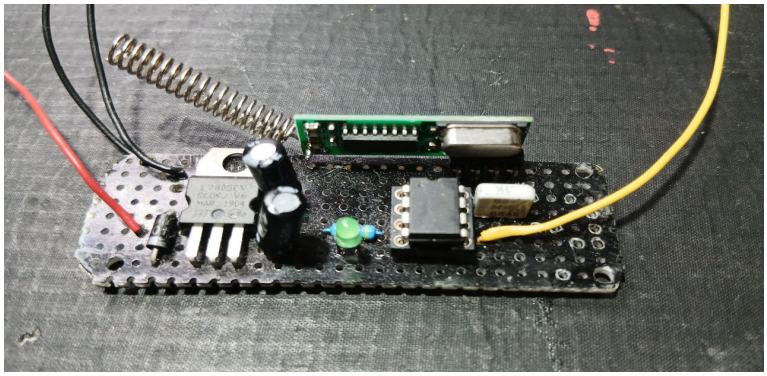


The receiver operates at 12 volts via a voltage regulator that reduces the voltage to 5 volts.
When the transmitter sends the string, the receiver lights the LED and triggers the relay.
In the actual receiver circuit, I didn't place the relay as shown in the diagram, simply because this is just a demo.

The receiver



Notes and improvements:
The receiver schematic includes a JP1 that I didn't use, but is intended for future use as a bistable configuration.
It can certainly be improved at the firmware level, but this is just an example that anyone can start from.
Caution: Remember the FUSE configuration, see the code.

```
/*****************************************************************
DRTEK
Name: ask-ook-tx-WL102
Microcontroller: ATTINY85
Date: 29/08/2025
Version: 00.01
Description: When the button is pressed, sends a string to the wl102-341 module, then enters sleep
mode until the next button press
Links and references
rev.:       date:       notes:
//!remember to configure the fuses in the programmer
Low Fuse: E2
High Fuse: DF
Extended Fuse: FF
*****************************************************************/
//!working phase
#chip tiny85,8
#option explicit

' ----- Definitions
#define En_tx portb.0       // enable tx  pin 5
#define tx    portb.1       // DATA pin 6
//#define led1  portb.4 not used
dir tx out

' ----- Global variables
dim bd as word
dim ix as byte
dim ch as byte
dim bit_index as byte
dim mask(8) as byte
dim msg(12) as byte     //! "test DRTEK\r\n" = 12 characters !!set the number of used characters
dim ii as byte
dim trigger as bit
dim prev_state as bit
dim zz as byte      //helper variable for message repetition loop
dim conta as byte
zz=0
' ----- Configure pin direction
dir En_tx out       //output for tx enable
dir portb.2 in      //button input pin 7
//set portb.2 on   // button input Pull-up on PB2 for energy saving

' ----- Initialization
set tx on             ' idle line
bd = 3333               //833      //speed
/*ASK/OOK speed
Baud   Rate Bit Duration (µs)   Note
300    3333                  Very slow, high tolerance
600    1666
1200    833                  Used in old modems
2400    416                  Good compromise
```

```
4800    208                 Faster, still stable
9600    104                 Standard TTL serial
*/
' ----- Bitmask table
mask(0) = 1
mask(1) = 2
mask(2) = 4
mask(3) = 8
mask(4) = 16
mask(5) = 32
mask(6) = 64
mask(7) = 128

' ----- Load ASCII message
msg(0) = asc("D")
msg(1) = asc("R")
msg(2) = asc("T")
msg(3) = asc("e")
msg(4) = asc("k")
msg(5) = asc("T")
msg(6) = asc("x")
msg(7) = asc("1")
msg(8) = asc("=")
msg(9) = asc("1")
msg(10) = 13          ' CR
msg(11) = 10          ' LF

//---------------------setup
-------------------------------------------------------------------------------------------------
  wait 100 ms
// ----- Initial blinking
       for ii = 1 to 10
          set EN_tx on    //led
          wait 10 ms
          set EN_tx off   //led
          wait 100 ms
       next ii

' ----- Disable unnecessary peripherals
PRR = 0b00001111   ' Turns off ADC, Timer0, Timer1, USI
ADCSRA = 0         ' Disables ADC

' ----- Configure interrupt on PB2 (PCINT2)
GIMSK = GIMSK or 0b00100000   ' Enable PCIE
PCMSK = PCMSK or 0b00000100   ' Enable PCINT2
SREG.7 = 1                    ' Enable global interrupts

On Interrupt PinChange0 Call wakeup

prev_state = portb.2   ' Initial pin state
//-----------------------end setup
-------------------------------------------------------------------------------------------------
```

```
' ----- Main loop
do
   MCUCR = MCUCR or 0b00110000   ' SM1:SM0 = 01 (Power-down)
   MCUCR = MCUCR or 0b00100000   ' SE = 1
   asm sleep                     //enter sleep mode

   if trigger = 1 then
      trigger = 0
 //     dir led1 out             //set led as output
 //     set led1 on
      set EN_tx on          //enable tx
      for ix = 0 to 2            //! ix must match number of characters to send
         ch = msg(ix)             //send =
         gosub sb               ' send byte
         wait 50 ms              //!do not change this timing
      next
      for zz=0 to 3 //repeat message 3 times
         for ix = 0 to 11            //! ix must match number of characters to send
            ch = msg(ix)
            gosub sb          ' send byte
            wait 10 ms
         next
         wait 10 ms
      next zz
      set EN_tx off         //disable tx
   //   set led1 off
   //   dir led1 in              //set led as input for low power
   end if

loop
//---------------routines and
functions-----------------------------------------------------------------------------------------
' ----- Routine to send a byte via ASK/OOK
sb:
   ' Start bit
   set tx off
   wait bd us

   ' 8 bits (LSB first)
   for bit_index = 0 to 7
      if (ch and mask(bit_index)) <> 0 then
         set tx on
      else
         set tx off
      end if
      wait bd us
   next

   ' Stop bit
   set tx on
   wait bd us
```

```
return
//-------------------------------------------------------------------------------------
sub wakeup
   if prev_state = 0 and portb.2 = 1 then
      trigger = 1
   end if
   prev_state = portb.2
end sub
//------------ end of program
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
/*****************************************************************
DRTEK
Name: ask-ook-rx-WL101-attiny85
Microcontroller: ATTINY85
Date: 29/08/2025
Version: 00.07
Description: Receives via ASK/OOK and turns on LED if it receives the string "DRTekTx1=1"
Notes: Baud rate 300, bit-banging decoding, string buffer
//!remember to configure the fuses in the programmer
Low Fuse: E2
High Fuse: DF
Extended Fuse: FF
//-------------------------------------------------
//!ASK/OOK speed
Baud   Rate Bit Duration (µs)   Note
300    3333
600    1666
1200    833
2400    416
4800    208
9600    104
14400   69
19200   52
//! status = working
*****************************************************************/

#chip tiny85,8
#option explicit
#include <SoftSerial.h>

' ----- UART serial configuration
#define SER1_BAUD 9600
#define SER1_DATABITS 8
#define SER1_STOPBITS 1
#define SER1_INVERT Off
#define SER1_TXPORT PORTB
#define SER1_TXPIN 1       ' pin 6
#define SER1_RXPORT PORTB
#define SER1_RXPIN 2       ' pin 7
#define SER1_RXNOWAIT Off
```

```
' ----- Hardware definitions
#define rx_pin pinb.0       ' ASK/OOK input (pin 5)
#define led1   portb.4      ' Status LED (pin 3)

dim BIT_DELAY as word
dim receivedByte as byte
dim value as byte
dim rxString as string

BIT_DELAY = 3333     ' Baud 300

' ----- Pin setup
dir led1 out
dir rx_pin in

' ----- Initial blinking
dim blink as byte
for blink = 0 to 3
   set led1 on
   wait 100 ms
   set led1 off
   wait 100 ms
next

' ----- Send test message over serial
Ser1Print "Ask-ook v0.1"
Ser1Send 13   ' CR
Ser1Send 10   ' LF

' ----- Main loop
do
   if waitForStartBit() then
      receivedByte = readByte()

      ' Build string only with printable characters
      if receivedByte >= 32 and receivedByte <= 126 then
         rxString = rxString + chr(receivedByte)
         /*  //debug
         Ser1Print "Char: " & chr(receivedByte) & " | String: " & rxString
         Ser1Send 13
         Ser1Send 10
         */
      end if

      ' If CR (13) or LF (10) is received, consider the string complete
      if receivedByte = 13 or receivedByte = 10 then
         if len(rxString) > 0 then

            if rxString = "DRTekTx1=1" then
               Ser1Print "Received: " & rxString
               Ser1Send 13
               Ser1Send 10
```

```
                set led1 on
                wait 500 ms
                set led1 off
            end if

            rxString = ""   ' reset
        end if
    end if
end if
loop

' ----- Function to detect start bit
function waitForStartBit() as bit
    dim timeoutCounter as byte
    timeoutCounter = 0

    do
        if rx_pin = 0 then
            wait BIT_DELAY / 2 us
            return true
        end if

        wait 1 ms
        timeoutCounter = timeoutCounter + 1
    loop until timeoutCounter > 5   ' timeout of about 5 ms

    return false
end function

' ----- Function to read a byte via ASK/OOK
function readByte() as byte
    dim ii as byte
    value = 0
    for ii = 0 to 7
        wait BIT_DELAY us
        if rx_pin = 1 then
            value = value or (1 << ii)
        end if
    next
    wait BIT_DELAY us   ' stop bit
    return value
end function
```