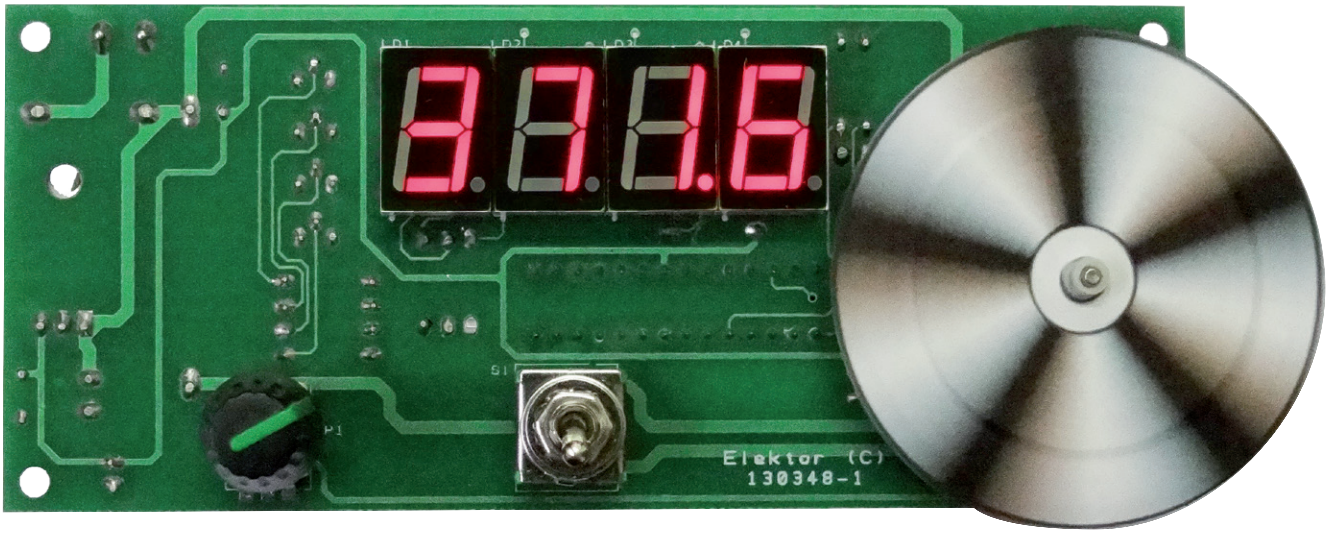
Coil Winder
It Winds and Counts

By
Rolf Gerstendorf
(Germany)



Many low frequency inductors used in long and medium wave receiver design require a few hundred turns of wire. Winding the coils by hand can be troublesome, especially if you lose count along the way. This simple coil winder is easy to build and spins the coil former clockwise or anticlockwise while keeping track of the turns and displaying the exact value with a resolution of one tenth of a turn.

This coil winder design, as the name suggests, is used to wind coils. It keeps track of the number of turns and the direction of rotation with the help of an encoder disc mounted on a motor shaft. The disc is divided into ten sections alternating black and white. Two opto-sensors are used to register the turn of the disc in either direction. Clockwise rotation of the motor shaft increments the value shown on an LED turns-counter display while counter-clockwise rotations decrement the value.

Hardware

The circuit diagram shown in **Figure 1** can be divided into five sections: First is the display consisting of four 7-segment LEDs. Next is a rotation impulse generator (both the display and impulse generator connect to the microcontroller). A DC motor controller and a voltage regulator.

Any intelligence in the operation of this coil winder is provided by the ATmega328P (IC2) microcontroller which is clocked from its internal 8-MHz clock signal. Its main function is to count rotations of the motor axle in both clockwise and counter-clockwise direction and to control the four 7-segment LEDs. The display can show a maximum of 999.9 turns. The circuit around the two reflective opto-sensors type CNY-70 (IC3 and IC4) generates the rotational turns count and direction of rotation for the controller. The opto-sensors are positioned next to each other where they illuminate the black and white encoder disc.

When light from the opto-sensors internal LED hits a white region of the disc it is reflected back to the phototransistor which conducts and pulls the output signal and the microcontroller pin to ground. The black region absorbs light so when this part of the disc is under the opto-sensor the phototransistor does not receive any reflected light and turns off. The corresponding controller input will now be pulled High by the pull-up resistors R13 or R14.

The disc has five white and five black segments which gives 10 changes of light level per revolution. The software monitors the change of input level from IC4 at pin PD2 and uses it to change the count value. At the same time it checks the output level of IC3 (pin PD6), if both inputs are at the same state the turns-counter is incremented, if they are different the turns counter is decremented. The display is made up of four 0.5 inch 7-segment LED displays with common cathodes. All seven segments from the displays LD1 to LD4 are connected in parallel to controller outputs PB0 to PB6. Pins PC0 to PC3 drive the common cathodes of the four display characters via transistor buffers T2 to T5. The decimal point pin on display LD3 is connected to the supply rail via resistor R12 so that the display has the format '000.0'.

The coil winder uses a N2738 12 V DC motor from Igarashi [1]. It weighs just 70 g and its low-cost makes it a favorite in the model-building community. The motor achieves maximum efficiency (66.7 %) with a load current of 1.43 A (11.4 W) running at 13,000 rpm. In this application we won't be getting anywhere near that speed (could lead

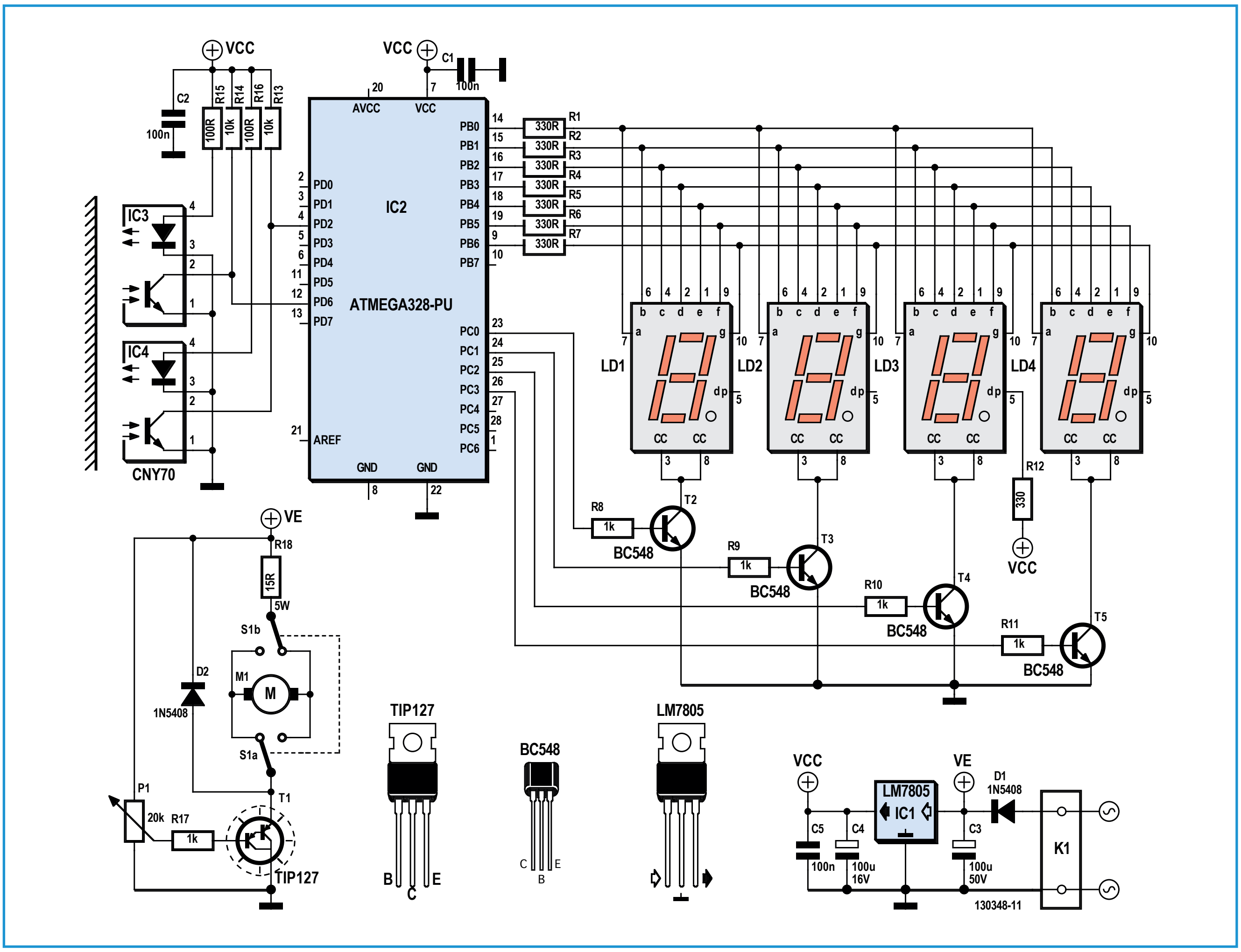


Figure 1.
The coil winder circuit.

to combustion of your digital extremities!), the technical data indicates that this motor can supply sufficient torque (8.37 mNm) for the application here.

Control of the motor speed is completely independent of the microcontroller and can be regarded as fairly ‘basic’. Power to the motor is supplied by the Darlington-transistor T1, the 20 K pot P1 simply controls the base current to T1 to change the motor speed. You just need to select the motor speed as required. For thicker wire use a slower motor speed. The motor can draw 10 A when it stalls so resistor R18 is used to limit the maximum current. The three position ‘ON-OFF-ON’ DPDT switch S1 is used to change the motor’s direction of rotation. The 1N5408 diode D2 is rated at 3 A to provide the freewheel function.

The circuit is powered from a 12 V wall adapter type power supply which connects to socket K1. Diode D1 protects the circuit from a reverse polarity power connection. The motor is powered directly from this supply while the rest of the circuit is powered from a stabilized 5 V supplied by the voltage regulator IC1.

Software

The software was written in C using AVRstudio. The program source file and the compiled version of the program can be downloaded for free from [2]. **Figure 2** shows the fuse settings used and how they are shown when programming in AVR-Studio.
The main program consists of two functions:

Display_seg-Function
This function takes the value of count as an argument, separates the value into individual numbers and displays them on the 7-segment LEDs.

ISR0-Routine
PD2 is used as an interrupt input. The interrupt service routine will be called at every status change of PD2. It compares the value on input PD6 with PD2 and increases the turns count (if PD6 is the same) or decrements (if PD6 is not the same) the count value.

Mainfunction (Hauptfunktion)
This is the main function which defines the I/O pin data direction, takes care of initialization, enables the interrupts and calls the Display_seg-Function.

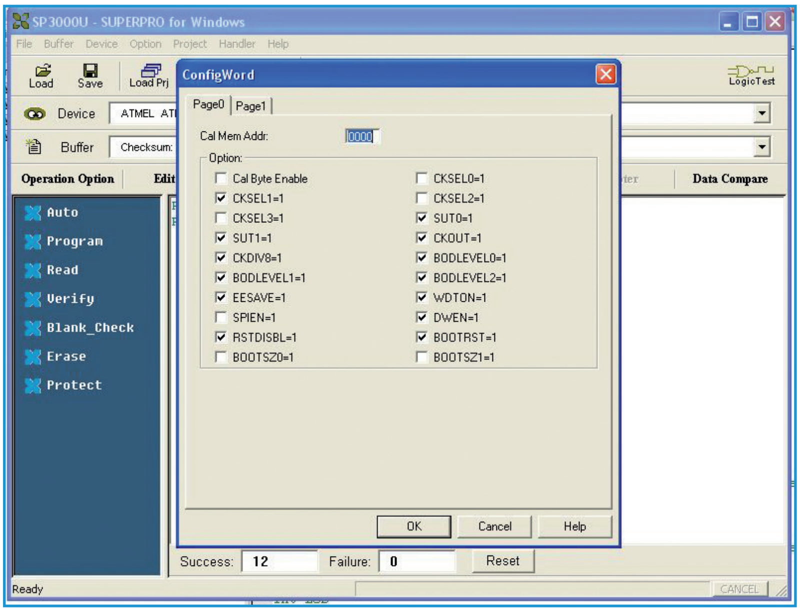


Figure 2.
The microcontroller fuse settings.

The Mechanics

All components, electrical and mechanical are fitted to the PCB (**Figure 3**), which was produced using the free PCB software called *Designspark PCB* [3]. The layout and software files can be found on the project page [2]. The PCB layout allows it to be fitted into a basic enclosure or project box. Holes are required to accommodate the pot P1 and switch S1. A cutout is also required for the four displays (fitted to the upper side of the PCB). A hole is also necessary in the light-tight enclosure for the motor shaft. Header sockets are fitted to the PCB to mount the 7-segment displays.

The two opto-sensors are also fitted to the upper side; all the other components are mounted on the reverse side of the double-sided PCB. Transistor T1 must be fitted with an effective heat sink.

The encoder disc is divided up into ten sections and then colored black and white. The easiest option is to make a photocopy of **Figure 4**; this will give you a correctly sized 50 mm diameter disc. It can be glued to a 50 mm diameter scrap piece of PCB. Ensure everything is symmetrical around the center hole to avoid vibrations when the motor

COMPONENT LIST

Resistors
R1–R7,R12 = 330Ω
R8–R11,R17 = 1kΩ
R13,R14 = 10kΩ
R15,R16 = 100Ω
R18 = 15Ω 5W
P1 = 20kΩ potentiometer

Capacitors
C1,C2,C5 = 100nF
C3 = 100µF 50V
C4 = 100µF 15V

Semiconductors
D1,D2 = 1N5408
LD1–LD4 = 7-Segment-display, 0.5", common cathode, e,g. Kingbright SC05-11EWA
T1 = TIP127
T2–T5 = BC548
IC1 = LM7805
IC2 = ATmega328-PU (Atmel), programmed, Elektor Store # 130348-41
IC3,IC4 = CNY70 (Vishay)

Miscellaneous
S1 = toggle switch, DPDT (TE Connectivity A203SYZQ04)
K1 = 2-way PCB screw terminal block, 5mm pitch
Pinheaders, 0.1'' pitch
12-V motor type N2738-051-G-5 RS385SH (Igarashi), Conrad Electronics # 244520
Propeller-Hub-Reel for 2.3mm spindle (10583), Conrad Electronics # 224235
PCB, Elektor # 130348-1 [2]

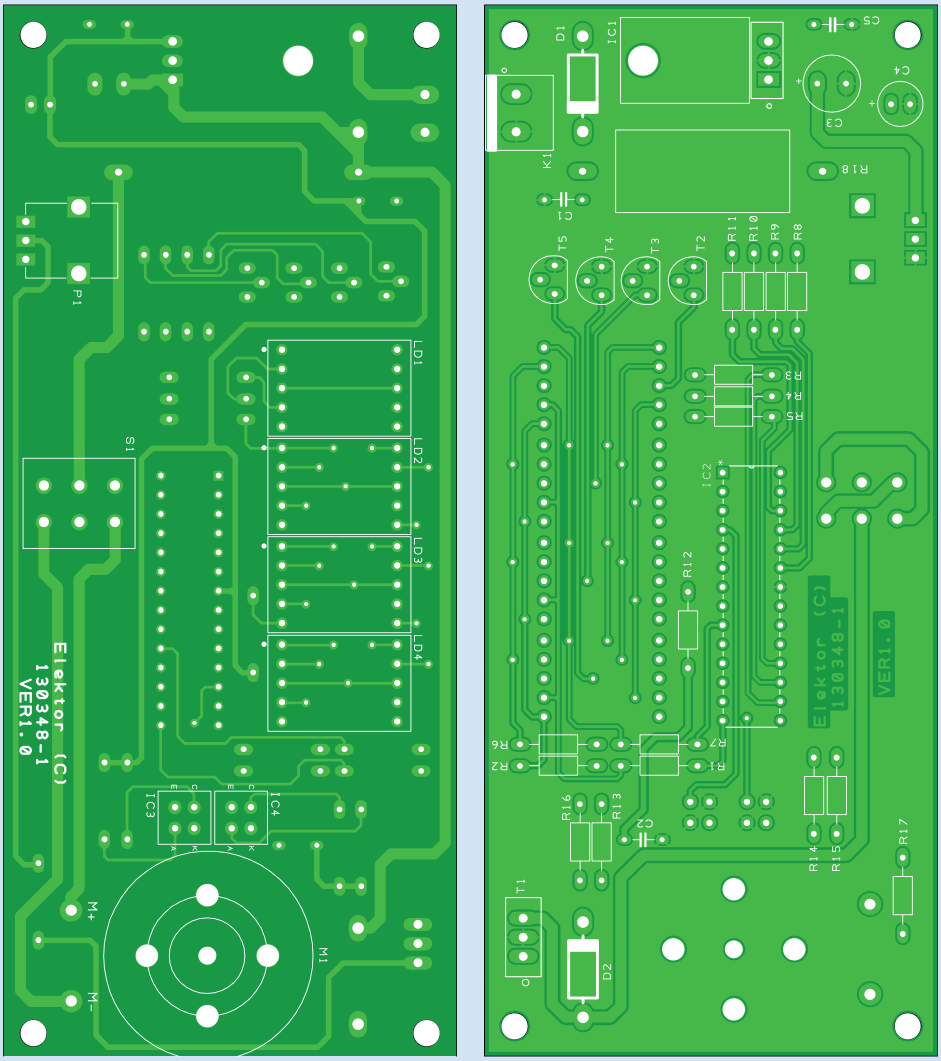


Figure 3.
All components, including the motor are fitted to the double-sided board.

spins up. The size of the central hole will be governed by the method used to attach it to the motor axle.

The motor itself is fixed to the PCB using two M3 x 0.5 x 5 mm screws as shown on the layout. Make sure that the screws don't extend any further than 3 mm into the motor housing. There are a number of ways that the disk and coil former can be fitted to the 2.3 mm diameter axle [4]. For this example we chose an electric flight propeller hub. The disc can be glued directly onto the shaft. Whichever method you use don't forget that the distance between the opto-sensor and the disc surface should be approximately 2 mm.
In the case where the motor is not fixed to the PCB (you could, for example use an electric

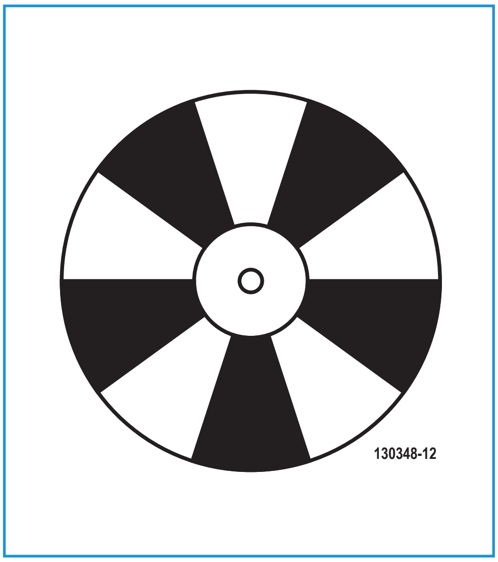


Figure 4.
1:1 template for the encoder disc.

drill to wind the coil) the opto-sensors will also need to be mounted off the PCB. However you configure your setup it is important to remember firstly, that a multi-core screened cable should be used for wiring between the opto-sensors and PCB. This helps reduce the chance of picking up interference signals in the cable which may lead to incorrect values displayed on the turns counter. Secondly, make sure that the opto-sensor is properly shielded from any external stray light.

(130348)

Web Links

[1] www.igusa.com/pages/motors/N2738.html

[2] www.elektor.com/en/post

[3] http://designshare.designspark.com/eng/projects/122/view/files

[4] www.robotmarketplace.com/products/hubs.html