

# Propelleruhr

## Mit ultra-stylishem Ziffernblatt

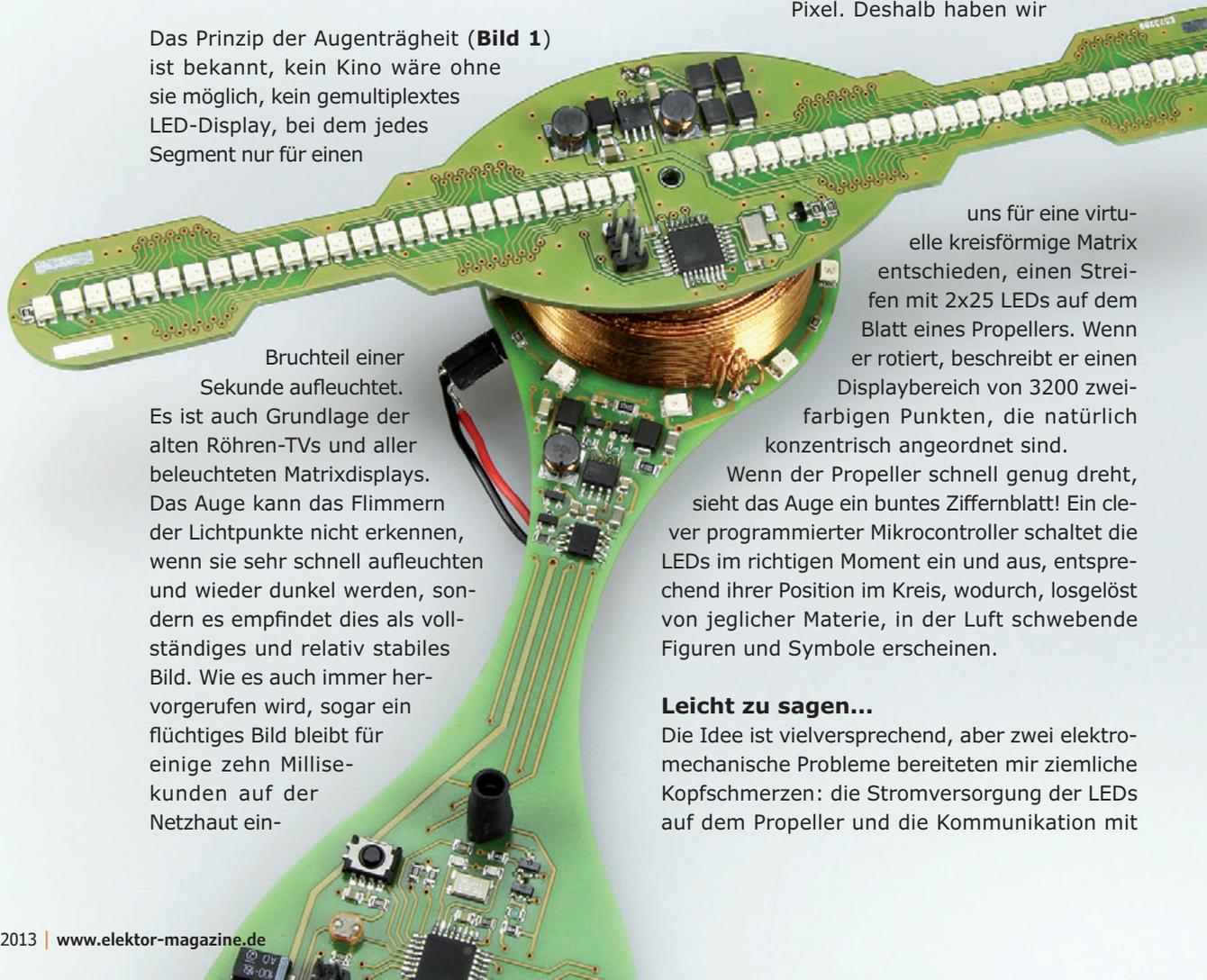
Elektronik ist am schönsten, wenn sie Physik, Mechanik und Software mit Phantasie, Ästhetik und Präzision kombiniert. Diese Uhr wurde zwar entwickelt, um Zeit und Datum anzuzeigen. Doch sie wurde auch entworfen, um Ausrufe der Bewunderung von Besuchern zu provozieren!

**von David Ardouin**  
(Frankreich)

Im Internet findet man mit Stichworten wie „Propelleruhr“ oder „mechanical sweep display“ eine Reihe von Projekten nach dem hier vorgestellten Prinzip, aber nicht viele von ihnen haben das Stadium des Laboraufbaus verlassen. Hier möchte ich Ihnen nicht nur zeigen, wie so etwas funktioniert, sondern wie man das Prinzip in eine hochwertige Uhr verwandelt, die Ihren Gästen erstaunte „Oohs“ und „Aahs“ entlockt.

Das Prinzip der Augenträgheit (**Bild 1**) ist bekannt, kein Kino wäre ohne sie möglich, kein gemultiplextes LED-Display, bei dem jedes Segment nur für einen

gestempelt. Alles, was man tun muss, ist, das Bild oft genug zu aktualisieren, um den Eindruck von Stabilität oder „glatter“ Bewegung zu vermitteln. Das Problem liegt hier eher in der Anzahl der Punkte. Je mehr es gibt, desto teurer wird die Hardware und desto schwieriger ist es, sie zu steuern. Bei einer statischen Displaymatrix steigen die Kosten schnell ins „Unermessliche“, sobald wir mehr zeigen wollen als nur ein paar Pixel. Deshalb haben wir



Bruchteil einer Sekunde aufleuchtet. Es ist auch Grundlage der alten Röhren-TVs und aller beleuchteten Matrixdisplays. Das Auge kann das Flimmern der Lichtpunkte nicht erkennen, wenn sie sehr schnell aufleuchten und wieder dunkel werden, sondern es empfindet dies als vollständiges und relativ stabiles Bild. Wie es auch immer hervorgerufen wird, sogar ein flüchtiges Bild bleibt für einige zehn Millisekunden auf der Netzhaut ein-

uns für eine virtuelle kreisförmige Matrix entschieden, einen Streifen mit 2x25 LEDs auf dem Blatt eines Propellers. Wenn er rotiert, beschreibt er einen Displaybereich von 3200 zweifarbigen Punkten, die natürlich konzentrisch angeordnet sind.

Wenn der Propeller schnell genug dreht, sieht das Auge ein buntes Ziffernblatt! Ein clever programmierter Mikrocontroller schaltet die LEDs im richtigen Moment ein und aus, entsprechend ihrer Position im Kreis, wodurch, losgelöst von jeglicher Materie, in der Luft schwebende Figuren und Symbole erscheinen.

### Leicht zu sagen...

Die Idee ist vielversprechend, aber zwei elektromechanische Probleme bereiteten mir ziemliche Kopfschmerzen: die Stromversorgung der LEDs auf dem Propeller und die Kommunikation mit

dem Mikrocontroller des Propellers. Eine Batterie würde nicht lange halten, und Schleifringe oder Bürsten kommen nicht in Frage, weil sie sehr schnell verschleiben und auch ziemlich laut sein würden. So entschied ich mich, die Induktion eines **speziell angefertigten Transformators** zu nutzen, um die Energie vom feststehenden zum rotierenden Teil zu übertragen.

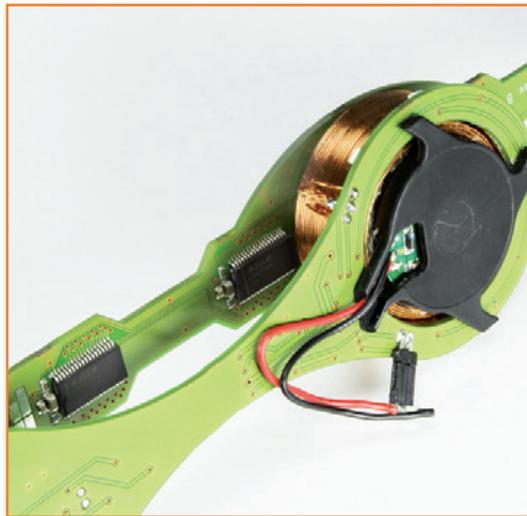
Die Wahl des Propellermotors ist von entscheidender Bedeutung, da er leise zu sein hat, schnell, einfach zu versorgen und so langlebig wie möglich. Offensichtlich eignen sich gewöhnliche alte Motoren mit Kohlebürsten nicht. Ein nahezu unhörbarer und strapazierfähiger bürstenloser Motor einer Festplatte wäre sehr verlockend, aber sein Metallgehäuse erwies sich als unvereinbar mit der Induktionsversorgung. So entschied ich mich für einen Computer-Lüfter, der einfach mit Spannung zu versorgen, dabei zuverlässig und ruhig, kräftig genug und aus nicht abschirmendem Kunststoff ist.

Es bleibt nun die heikle Frage der Kommunikation mit dem Controller. Alle Bedienelemente befinden sich auf der stationären Basiseinheit, am Propeller sind nur die Teile angebracht, die unbedingt für die Ansteuerung der LEDs notwendig sind. Nur so konnte die bewegte Masse reduziert und damit übermäßige Vibration, Abnutzung und Lärm vermieden werden. Um drahtlos zwischen Basiseinheit und Propeller zu kommunizieren, gebrauche ich eine Infrarot-Verbindung, die von einem festen Ring von IR-Sendediode zu einem Photodetektor auf dem Propellerblatt reicht. Nach vielen vergeblichen Versuchen und mehreren Generationen von Prototypen mündete dies in einer Elektronik, deren Blockschaltung in **Bild 2** zu sehen ist. Die beiden wichtigsten Komponenten in diesem Projekt sind in der Mitte des Diagramms zu sehen: Tr1, der auf dem Motor M sitzt. Über den Trafo werden die LEDs und ihre Treiberschaltung versorgt. Sie nehmen sich gegenüber der Basiseinheit oder der LED-Ansteuerung unscheinbar aus, aber wir werden dennoch sehr viel über sie reden.

### Ein magischer Propeller

Ich werde mit dem zweiblättrigen Propeller beginnen, da dies zweifellos der faszinierendste Teil der Geschichte ist. Der Algorithmus, nach dem dieses „schwere lose“ Display arbeitet, ist erstaunlich einfach. Der Propeller ermittelt seine Winkelposition durch einen Fototransistor am Ende eines

Blatts. Dieser wird bei jeder Umdrehung von einer Infrarot-LED auf der Basiseinheit beleuchtet. Auf diese Weise erhält der Mikrocontroller auf dem Blatt einen kurzen Impuls bei jeder Umdrehung. Ein interner Zähler misst die Zeit zwischen zwei Impulsen, also die Dauer einer Umdrehung. Der gelesene Wert wird durch 128 geteilt, entsprechend der Zahl der „Speichen“ des kreisförmigen Bildes. Dieses Ergebnis wird dann einem zweiten Zähler zugeführt, der den Programmablauf 128 Mal pro Umdrehung unterbricht. Eine Matrix von 128 Bytes, in der jedes Bit dem Zustand einer LED entspricht, wäre ausreichend, um den Zustand von 8 LEDs an jeder Stelle zu speichern.



So viel zur Theorie. Klingt einfach, nicht wahr? Alles wird mit den wenigen Programmzeilen in den Interrupts erledigt und dies bei nur geringen Anforderungen an die Prozessor-Ressourcen. Dank dieser ständigen Messung der tatsächlichen Dauer einer Umdrehung bleibt die Anzeige spektakulär stabil, unabhängig von der Drehzahl.

In der Praxis ist es doch schon ein wenig komplizierter, aus drei Gründen. Erstens hat jede Spalte zwei Farben und wird gleich von 25 LEDs gebildet (**Bild 3a**), so dass die Matrix nicht 128, sondern 768 Bytes zählt. Zweitens berechnet, um eine Stabilität des Displays zu erreichen, der Prozessor die Dauer der aktuellen Umdrehung aus der Beschleunigung des Propellers, der er wiederum durch einen Vergleich der beiden vorangegangenen Umdrehungen ableitet. Drittens hatte ich aus den Experimenten gelernt, dass ich unbedingt ein Flackern des Displays vermeiden



Bild 1. Die Augenträgheit ermöglicht Bilder wie diese.

## So funktioniert's:

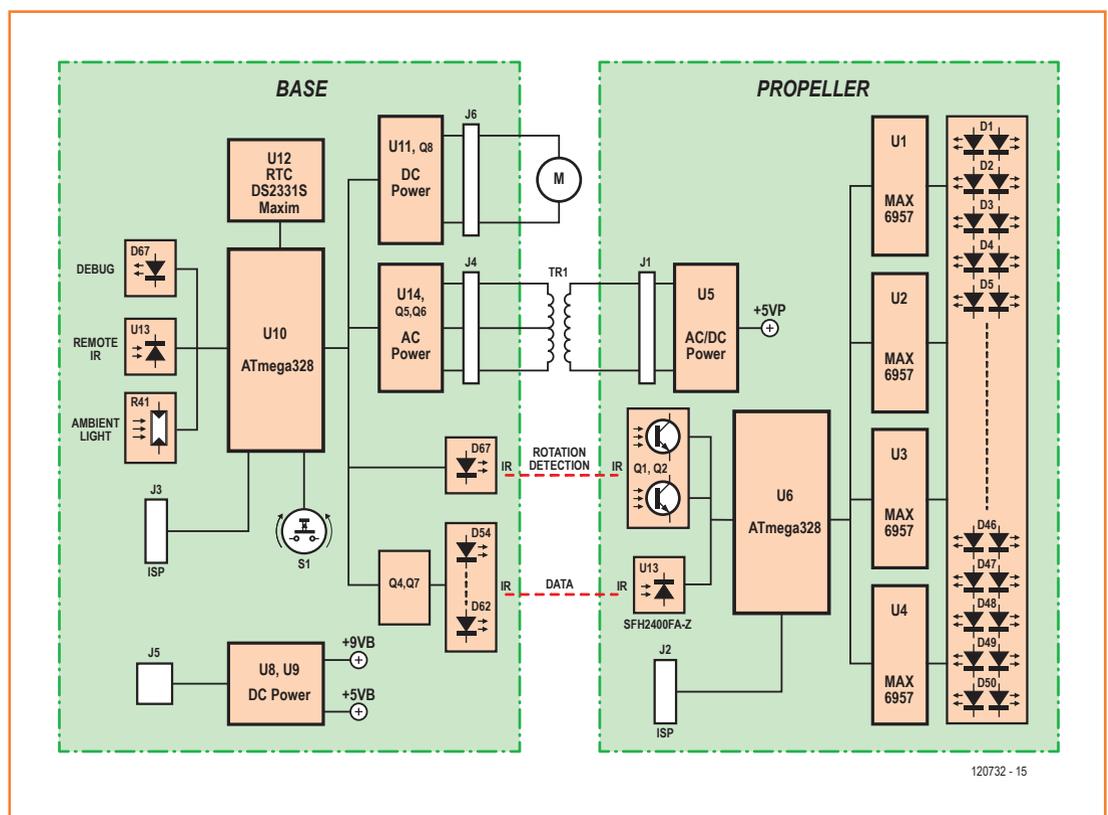
1. Die Uhr verfügt über zwei ATmega328-Mikrocontroller, einer in der festen Basisstation und der andere auf dem Zwei-Blatt-Propeller.
2. Der Propeller-Mikrocontroller steuert 2x25 LEDs und zeigt dank der Augenträgheit ein kreisförmiges Bild mit 3200 Punkten.
3. Der Propeller ist auf die Nabe eines Lüftermotors geklebt und dreht sich mit ihm.
4. Der Stator dieses Motors ist mit der Basis-Einheit verklebt.
5. Die elektrische Energie wird von der Basis zum Propeller drahtlos übertragen. Der Transformator besteht aus zwei konzentrischen Windungen um den Motor, dessen Lüfterflügel abgeschnitten wurden.
6. Die Transformator-Primärwicklung (Außenwicklung) ist mit der Basis verklebt, die sekundäre (Innenwicklung) mit der Motornabe.
7. Die Displaydaten an den Propeller-Mikrocontroller werden von einem Ring von IR-Sendern auf der Basis-Einheit durch ein Infrarot-Signal zu einer sich darüber drehenden Photodiode übermittelt.
8. Der Propeller ist unhörbar, da er relativ langsam dreht. Das Bild, das durch die Augenträgheit entsteht, ist stabil, weil einerseits die beiden Blätter abwechselnd das gleiche Fragment erzeugen und andererseits der Mikrocontroller das Bild der Drehzahl des Propellers anpasst.
9. Praktisch alle Aufgaben werden von vier Interrupts gesteuert.
10. Ein einziger Drehencoder mit Taster wird an der Basiseinheit verwendet, um alle Einstellungen vorzunehmen: Standby, Zeit- und Datum-Einstellung, Helligkeit und Drehzahl zur Tag- und Nachtzeit, Sprachauswahl und Display-Modus (61 mögliche Konfigurationen!).

musste. Dazu wäre eigentlich eine ziemlich hohe Geschwindigkeit erforderlich, aber eine höhere Geschwindigkeit bedeutet auch mehr Lärm. Also habe ich die Anzeige verdoppelt: Beide Blätter des Propellers emittieren dieselben Lichtsignale, allerdings mit einer Verzögerung von einer halben Umdrehung, also 64 Speicher. Je stabiler das Display, desto angenehmer ist es für das Auge, auch bei einer relativ niedrigen Geschwindigkeit in der Größenordnung von 1500 Upm.

Bei dieser Geschwindigkeit ist der Propeller fast unhörbar!

Die Hauptaufgabe des Propeller-Mikrocontrollers ist es, die Punktmatrix mit stets anderen Daten zu füllen. So muss der Controller die auf der seriellen Schnittstelle über die Infrarot-Verbindung empfangenen Sequenzen decodieren und die Display-Tabelle mit den Daten für eine der beiden möglichen Schriften (5x7 oder 6x10 Pixel) füllen. Er erledigt auch das Umschalten der Zei-

Bild 2. Blockschaltbild der Uhr, aufgeteilt auf die beiden Platinen für Basisgerät und Propeller. Dazwischen die drei wesentlichen Mechanismen Motor, Transformator und der (unsichtbare, aber unverzichtbare) Infrarot-Link.



120732 - 15



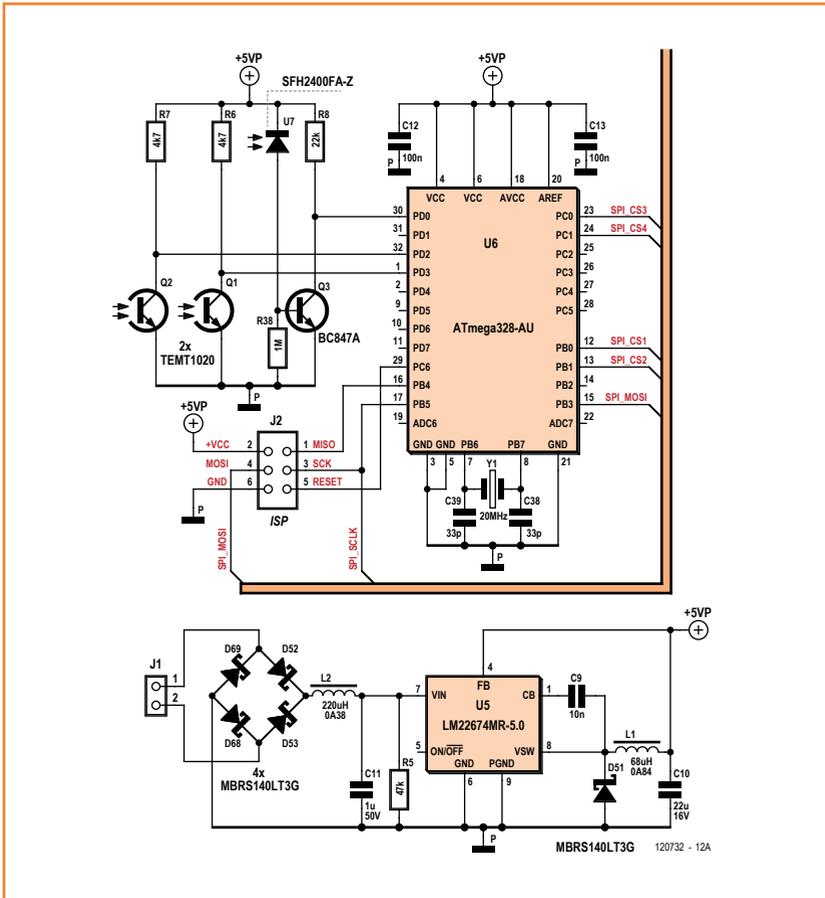


Bild 3b. Die aktive Hälfte der Propellerschaltung mit der Stromversorgung des Mikrocontrollers und des Fotodetektors.

chen, abhängig davon, ob sie sich im oberen oder unteren Halbkreis des Displays befinden. Der Mikrocontroller steuert auch die Helligkeit der LEDs und erzeugt die Zeiger der Uhr im analogen Modus zusammen mit der Sekundenanzeige (Bild 1).

## Elektronik für den Propeller

Die 25 LEDs pro Propellerblatt machen es möglich, drei Zeilen mit acht Pixel hohen Zeichen anzuzeigen. Die letzte dient der Sekundenanzeige am äußeren Rand des Kreises. Ich habe mich für rot/weiße zweifarbige LEDs entschieden. Die MAX6957 verwenden Konstantstromquellen, so dass die Nennarbeitsspannung dieser LEDs unwichtig ist. Damit sind Sie in der Lage, andere LED-Typen zu wählen mit Farben, die Sie lieber mögen, solange die Anschlussbelegung kompatibel ist. Die Treiber U1...U4 sind sehr handlich, sie steuern bis zu 28 Ausgänge mit Hilfe eines einfachen SPI (Serial Peripheral Interface). Der Strom für jeden Ausgang kann auf bis zu 20 mA eingestellt werden, perfekt für die Steuerung der Helligkeit der Anzeige. Mit modernen High-Bright-

ness-LEDs (über 100 mcd) ist die Leistung mehr als ausreichend für den Innenbereich.

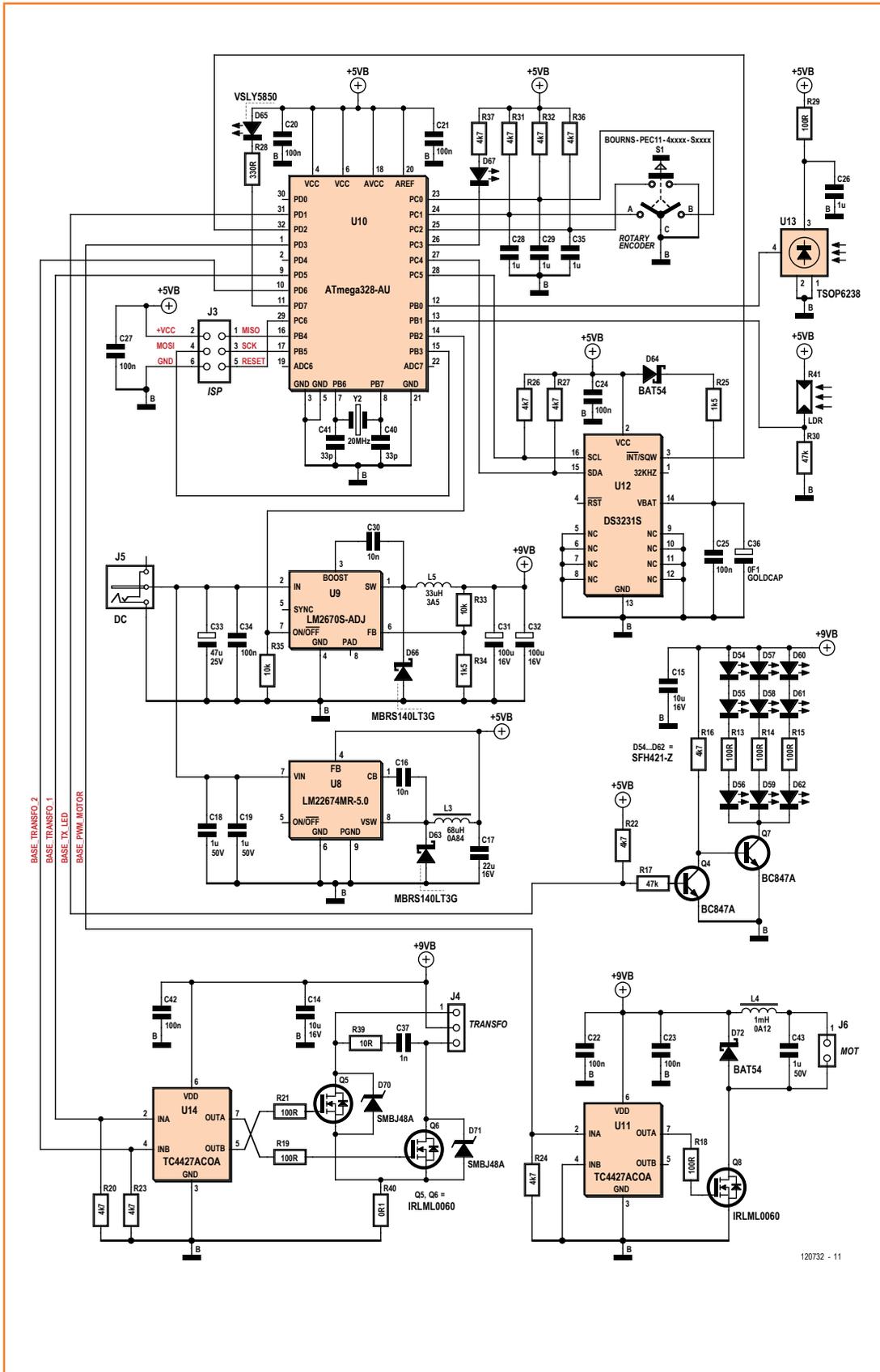
Als Mikrocontroller habe ich den ATmega328 (Bild 3b) gewählt, vor allem, weil er mindestens 1 KB RAM zur Speicherung der Display-Matrix bereitstellt. Er wird mit 20 MHz getaktet, um eine so schnell wie mögliche SPI-Verbindung zum Aktualisieren der LEDs zu realisieren.

Damit die Infrarot-Verbindung zwischen Basis und Propeller während der gesamten 360°-Drehung des Propellers aufrechterhalten werden kann, habe ich eine schnelle Weitwinkel-Photodiode (SFH2400) gewählt. Die FA-Version dieser Diode bietet auch eine Filterung des sichtbaren Lichts, das kann auf keinen Fall schaden. Beleuchtet durch eine Infrarot-Lichtquelle liefert diese Diode einen Strom von einigen Mikroampere, der verstärkt und von Q3 invertiert wird. Damit wird ein Ausgangspegel erreicht, der direkt kompatibel mit dem RX-Eingang des USART U6 ist.

Die Winkelstellung des Propellers wird durch die Fototransistoren Q1 und Q2 am Ende jedes Blattes ermittelt, jedes Mal, wenn sie die fixe LED passieren. Die Versorgungsspannung bezieht die Propellerelektronik von der Sekundärwicklung des Transformators Tr1 (dazu später mehr) über J1. Die vier Dioden D52, D53, D68 und D69 stellen einen gewöhnlichen Brückengleichrichter dar, L2 und C11 ein Filter. Im Leerlauf liegt die Spannung hier in der Größenordnung von 15 V, U5 regelt diese Spannung auf +5 V für die folgenden Bauteile herunter.

## Software für den Propeller

Die Funktionsweise der Software wird verständlich, sobald man das Prinzip des „fließenden“ Displays begriffen hat. Nach der Initialisierung der Peripherie (intern und SPI) geht das Programm in den Wait-Zustand. Alles wird von vier Interrupts gesteuert. Wenn eine serielle Sequenz empfangen wird, speichert die Software die empfangenen Bytes, decodiert sie und führt die entsprechenden Aktionen aus (Füllen der Matrix mit neuen Werten, Aktualisierung der Zeit oder Einstellen des gesamten LED-Stroms). Während eines externen Interrupts (von Q1 oder Q2 ausgelöst) wird der Wert im Counter 1 gelesen - dies entspricht der Dauer einer Umdrehung. Geteilt durch 128 wird dieser Wert in den Counter 0 geladen und der Display-Zeiger neu initialisiert. Erreicht der Zähler das Maximum, ohne zurückgesetzt zu werden, ist



120732 - 11

Bild 4. Schaltung der Basiseinheit. Der Motor und der Transformator sind nur in Bild 2 eingezeichnet.

## Listing 1. Kommunikationsprotokoll zwischen Basiseinheit und Propeller

```
Command : [DISPLAY_TYPE] : 0x00
Data    : [ANA/NUM, DATEEN, TEMPEN, MDCOL, MDEN, SECCOL, SECRNG1, SECRNG0] :
- ANANUM : Display format [0: Analog, 1: Numeric]
- DATEEN : Display date [0: Disabled, 1: Enabled]
- TEMPEN : Display temperature [0: Disabled, 1: Enabled]
- MDCOL  : Minutes dots color [0: Color 1, 1: Color 2]
- MDEN   : Display minutes dots [0: Disabled, 1: Enabled]
- SECCOL : Seconds ring color [0: Color 1, 1: Color 2]
- SECRNG : Seconds ring type [00: Disabled, 01: Elapsed Seconds, 10: Full ring, 11: Fixed ring]

Command : [LUM_POWER] : 0x01
Data    : [UNUSED, UNUSED, UNUSED, UNUSED, LUMPWR3, LUMPWR2, LUMPWR1, LUMPWR0]
- UNUSED : Unused [0000]
- LUMPWR : Luminous Intensity [0000: Minimum to 1111: Full power]

Command : [TIME] : 0x02
Data    : [HMCOL, UNUSED, UNUSED, HOURS4, HOURS3, HOURS2, HOURS1, HOURS0]
- HMCOL : Hands color (Analog mode), Hours text color (numeric mode) [0: Color 1, 1: Color 2]
- UNUSED : Unused [00]
- HOURS : Current hour [0x00 to 0x17]
[UNUSED, UNUSED, MINUT5, MINUT4, MINUT3, MINUT2, MINUT1, MINUT0]
- UNUSED : Unused [00]
- MINUT : Current minute [0x00 to 0x3B]
[UNUSED, UNUSED, SECON5, SECON4, SECON3, SECON2, SECON1, SECON0]
- UNUSED : Unused [00]
- SECON : Current second [0x00 to 0x3B]

Command : [DATE] : 0x03
Data    : [DATCOL, UNUSED, LANG1, LANG0, MONTH3, MONTH2, MONTH1, MONTH0]
- DATCOL : Date text color [0: Color 1, 1: Color 2]
- UNUSED : Unused [0]
- LANG   : Display language [00: Eng, 01: Fr, 10: Ger, 11: Undefined]
- MONTH : Current month [0x00 to 0x0B]
[DAYWK2, DAYWK1, DAYWK0, DATE4, DATE3, DATE2, DATE1, DATE0]
- DAYWK : Day of week [000: Monday to 110: Sunday]
- DATE  : Current date [0x00 to 0x1E]

Command : [TEMPERATURE] : 0x04
Data    : [TEMPCOL, UNUSED, TEMP6, TEMP5, TEMP4, TEMP3, TEMP2, TEMP1]
- TEMPCOL : Temperature text color [0: Color 1, 1: Color 2]
- UNUSED : Unused [0]
- TEMP : Integer portion of temperature
[TEMPFRAC1, TEMPFRAC0, UNUSED, UNUSED, UNUSED, UNUSED, UNUSED, UNUSED]
- TEMPFRAC: Fractional portion of temperature
- UNUSED : Unused [000000]

Command : [DISPLAY_TEXT] : 0x05
Data    : [TXTCOL, TXTCLR, TXTSIZE, UNUSED, UNUSED, UNUSED, SECT1, SECT0]
- TXTCOL : Text color [0: Color 1, 1: Color 2]
- TXTCLR : Clear sector prior to write new data [0: Keep previous text, 1: Clear then write]
- TXTSIZE : Text font [0: FONT_6x7, 1: FONT_8x16]
- UNUSED : Unused [000]
- SECT   : Sector number [0x00 to 0x03]
[UNUSED, TXT6, TXT5, TXT4, TXT3, TXT2, TXT1, TXT0]
- UNUSED : Unused [0]
- TXT    : ASCII Character to display[0x30 to 0x3A, 0x41 to 0x5A, 0x61 to 0x7A]
[TXT]
- Max 10 ASCII Characters in FONT_6x7 or 8 chars in FONT_8x16

Command : [TEST_FRAME] : 0x06
No Data

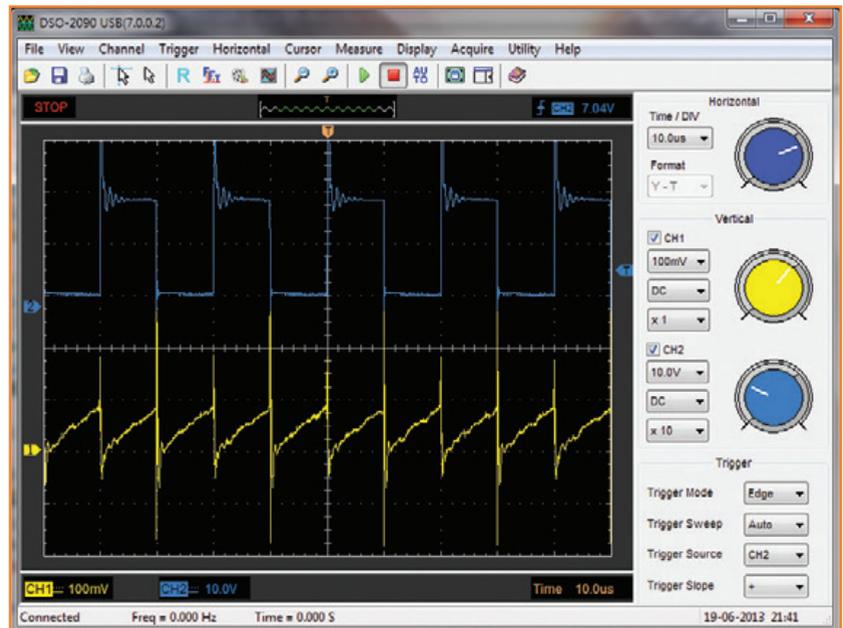
Command : [CHRISTMAS_TREE] : 0x07
Data    : [UNUSED, UNUSED, UNUSED, UNUSED, UNUSED, UNUSED, UNUSED, TREEEN]
- UNUSED : Unused [00000000]
- TREEEN : Christmas Tree [0: Disabled, 1: Enabled]
```

die Drehzahl zu niedrig, so dass die LEDs deaktiviert werden. Andernfalls, während des periodischen Interrupts durch Counter 0 (also 128 Mal pro Umdrehung), wird der Pointer der Displaymatrix inkrementiert und der entsprechende Wert an die LED-Ansteuerschaltungen gesendet. Wenn die Zeit im Analog-Modus angezeigt werden soll, übernimmt die Software auch die Darstellung der Zeiger und des Sekundenpunkts am Rand (siehe **Listing 2** mit dem Pseudo-Code für den Propeller).

### Drahtlose Energieübertragung

Die grundlegende Funktion der Basiseinheit (**Bild 4**) ist es, die Energie auf den Propeller zu übertragen. Der speziell gewickelte Transformator um die Motornabe ist konzentrisch zur Drehachse. Die primäre Windung auf der Außenseite ist fix an der Basiseinheit befestigt und bewegt sich nicht. Die sekundäre mit einem etwas kleineren Durchmesser ist in der Mitte der primären um den Motor positioniert, dreht sich also mit dem Propeller. Die Wechselspannung, die an die Primärwicklung über U14, Q5 und Q6 gelegt wird, bewirkt einen magnetischen Fluss konzentrisch zur Drehachse, der wiederum eine alternierende Spannung in der Sekundärspule induziert. Der Transformator ist in einer Gegentakt-Konfiguration geschaltet: Die Primärwicklung besteht aus zwei Hälften, die an Q5 und Q6 angeschlossen sind, die abwechselnd von einem Rechtecksignal mit einer Frequenz von 50 kHz von den Mikrocontroller-Ausgängen Counter 0, OC0A und OC0B angesteuert werden.

Dank des Windungsverhältnisses von 1:1,73 erhalten wir auf der Sekundärseite eine Rechteckspannung zwischen -15 V und +15 V, die gleichgerichtet, geglättet und auf +5 V heruntergeregelt wird. Die Bedämpfungsschaltung R39 und C37, unterstützt von den „Transzorb“-Dioden D70 und D71 (eine Art sehr schneller Zenerdioden), begrenzt Spannungsspitzen an den Transistoranschlüssen beim Schalten. Um ein sauberes Schalten zu gewährleisten, stellt Treiber U14 den Transistoren einen Strom zur Verfügung, der deutlich höher ausfällt als dies bei einem einfachen Mikrocontroller-Ausgang der Fall wäre. Über Widerstand R40 wird der Strom abgebildet (**Bild 5**), man kann hier eine dreieckige Spannungsform feststellen. Die Amplitude erreicht knapp 100 mV, was etwas weniger als 1 A entspricht. In diesem Konverter gibt es zu jedem



Zeitpunkt immer einen leitenden Transistor. Wenn diese Bedingung nicht erfüllt wäre, würde der Strom absinken und schließlich „ausbrennen“. Es ist also sehr wichtig, dass Sie das Programm nicht stoppen, zum Beispiel durch das Anschließen eines Programmier-/Debugging-Tools. Wenn Sie dies dennoch wagen möchten, stellen Sie sicher, dass Ihr Tool die Counter auch im Pause-Zustand weiter laufen lässt, sonst dürfte der erste Haltepunkt „tödlich“ sein. Diese Option ist im Konfigurationsmenü des Debugging-Tools unter AVR Studio 4 oder Atmel Studio 6 verfügbar.

### Der Propeller dreht sich

Jetzt haben wir die Mittel einer drahtlosen Energieversorgung zum Propeller, nur muss er auch schnell genug drehen, um die erhoffte Wirkung zu erzielen. Kraft und Geschwindigkeit eines PC-Lüftermotors sind perfekt für die zu bewegende Masse. Leise, beinahe unhörbar ist das Ding! Und all dies für so wenig Geld! Die Drehzahl wird über einen anderen PWM-Ausgang von U10, Treiber U11 und Schalttransistor Q8 eingestellt. Die PWM-Frequenz beträgt ebenfalls 50 kHz. Zusammen mit D72, L4 und C43 stellt dies einen Step-down-Konverter dar, der, abhängig vom Tastverhältnis des PWM, den Motor mit einer Spannung von 0...9 V versorgt. Wenn Sie vertraut sind mit dieser Art von Stromversorgung, werden Sie die ungewöhnliche Position des Transistors bemerken: Er ist massebezogen und auf

Bild 5. Spannung auf einer der Primärwicklungen (blau) des Transformators TR1 und über R40 (gelb).

## Listing 2. Pseudocode für die Propelleruhr

```
void main(void)
{
  do
  {
    switch(stateMachine)
    {
      case INIT:
      {
        //Initialise variables, microcontroller's registers, interrupts, LED drivers, USART
        stateMachine = IDLE;
      }
      case IDLE:
      {
        //All time critical operations occur in external and timer interrupts
        //New serial frame received ?
        if (g_u8_frameReceived == 1)
          //Process incoming frame : Decode, fill new display matrix, update hands position
        }
      }
    }
  }
  while(1);
}

//Interrupt called once per revolution
#pragma vector = INT1_vect
__interrupt void MCU_Int1Interrupt(void)
{
  //Set column index for hand 0 at 32+64
  u8_columnIndexHand0 = 32;
  //read Timer 1 value
  u16_revolutionPeriod = TCNT1;
  //divide Timer 1 value by 128, and set Timer 0 period with this value
  OCR0A = (u8)((u16_revolutionPeriod >> 7)-1);
  //reset and enable timer 1
  MCU_EnableTimer1();
  //Update display variable
  MCU_UpdateColumn();
  //Update LED status
  LED_Update();
}

//Interrupt called 128 times per revolution
#pragma vector = TIMER0_COMPA_vect
__interrupt void MCU_ColumnInterrupt(void)
{
  //Increment index
  u8_columnIndexHand0++;
  //Update display variable
  MCU_UpdateColumn();
  //Update LED status
  LED_Update();
}

//This function updates display bytes from display matrix
void MCU_UpdateColumn(void)
{
  //Update display variable for hand 0
  g_currentColumn.hand0OuterWhite = g_u8_displayOuterWhite[u8_columnIndexHand0];
  g_currentColumn.hand0OuterRed = g_u8_displayOuterRed[u8_columnIndexHand0];
  g_currentColumn.hand0MiddleWhite = g_u8_displayMiddleWhite[u8_columnIndexHand0];
  g_currentColumn.hand0MiddleRed = g_u8_displayMiddleRed[u8_columnIndexHand0];
  g_currentColumn.hand0InnerWhite = g_u8_displayInnerWhite[u8_columnIndexHand0];
  g_currentColumn.hand0InnerRed = g_u8_displayInnerRed[u8_columnIndexHand0];
}
```

diese Weise einfacher anzusteuern. Allerdings liegt der Ausgang nicht mehr direkt an der negativen Spannung, aber das ist in diesem besonderen Fall kein Problem.

**Der Propeller kommuniziert**

Nun haben wir nur noch die Aufgabe, die Daten drahtlos zu übertragen, die im Propellerdisplay angezeigt werden sollen. Im rotierenden Teil befindet sich eine IR-Fotodiode, die immer auf den auf der Basiseinheit montierten Ring der neun IR-LEDs D54...D62 mit breitem Abstrahlwinkel schaut. Diese LEDs werden gemeinsam versorgt und vom UART-Ausgang des ATmegas moduliert. Der nötige Strom durch die LEDs fließt über Schalttransistor Q7. Wegen der Invertierung durch Q4 entspricht ein logisches High am TX-Anschluss dem „dunklen“ Zustand der LEDs. Die Wahl von schnellen Dioden zum Senden und Empfangen gewährleistet die erforderliche Datenrate von 19.200 Baud. Bei dieser Geschwindigkeit ist die Zuverlässigkeit der Kommunikation beeindruckend. Die Paketierung der übermittelten Daten bietet eine bessere Immunität gegen alle kurzen Störungen des Kommunikationskanals. Jede Sequenz beginnt mit dem ASCII-Zeichen DLE (Data Link Escape, 0x10), gefolgt von einem STX (Start of Text, 0x02). Dann kommen maximal 66 Nutzbytes. Die Paketsequenz endet mit einem zweiten DLE-Byte, gefolgt von einem ETX (End of Text, 0x03). Nun könnte ein 0x10-Byte innerhalb der Nutzdaten enthalten sein. Um Verwechslungen zu vermeiden, wird das Byte automatisch zwei Mal gesendet und entsprechend beim Empfang behandelt. Der Empfänger bewertet nur komplette Sequenzen (**Bild 6**).

Das Kommunikationsprotokoll für diese Bytes ist spezifisch für den Propeller. Eine vollständige Liste der verfügbaren Befehle ist in **Listung 1** angegeben. Jede übertragene Sequenz beginnt mit dem Paar DLE/STX, dann weiter mit einem Befehlsbyte, das zwischen 0 und 5 liegt. Die folgenden Bytes sind die Parameter für den Befehl, deren Länge variieren kann. Schließlich endet die Sequenz mit dem Paar DLE/ETX, was die Verarbeitung der empfangenen Daten auslöst. Für die Anzeige von Text wird das Anzeigefeld in vier Sektoren (**Bild 7**) unterteilt. Bis zu zehn alphanumerische Zeichen können in jeder Zone in der kleinsten Schriftgröße angezeigt werden. Ein zweiter Satz von Zeichen, etwas größer, ergibt ein Maximum von acht Symbolen. In diesem Fall

DLE	STX	'A'	'B'	'C'	'D'	'E'	DLE	ETX
0x10	0x02	0x41	0x42	0x43	0x44	0x45	0x10	0x03

sind nur die Sektoren Null und Eins verfügbar. Alle Symbole werden durch Bytes codiert, bei der ein gesetztes Bit einem Aufleuchten einer LED entspricht, Bit 0 liegt am oberen Rand des Zeichens. Im unteren Halbkreis der Anzeige stellt die Software die Symbole um 180° gedreht dar. Das Bit 0 der Symbolcodierung bestimmt hier nicht mehr das Aufleuchten einer LED am äußeren Rand, sondern am inneren Rand des leuchtenden Zeichens.

Bild 6. Beispiel für die Sequenz „ABCDE“.

**Mikrocontroller und Peripherie**

Genau wie beim Propeller arbeitet auch in der Basiseinheit ein Mikrocontroller des Typs ATmega328. Zwei Drittel des 32 kB großen Programmspeichers sind noch frei (für zukünftige Anwendungen...). Um die Helligkeit des Displays einzustellen, wird das Umgebungslicht von LDR R41 erfasst. Die Infrarot-LED D65 strahlt ihr Licht in einem sehr schmalen Winkel ab und ist so hervorragend für den Positionsdetektor des Propellers geeignet. Wird diese LED deaktiviert, erlischt das Propeller-Display unmittelbar. S1 ist ein Drehencoder mit integriertem Taster, was die gesamte Mensch/Maschine-Schnittstelle darstellt, einschließlich der Navigation in einem Konfigurationsmenü und der Anpassungen der Einstellungen. D67 ist eine einfache orangefarbene LED, die im Sekundentakt blinkt. Sie können diese auch weglassen, wenn das Geblinke stört.

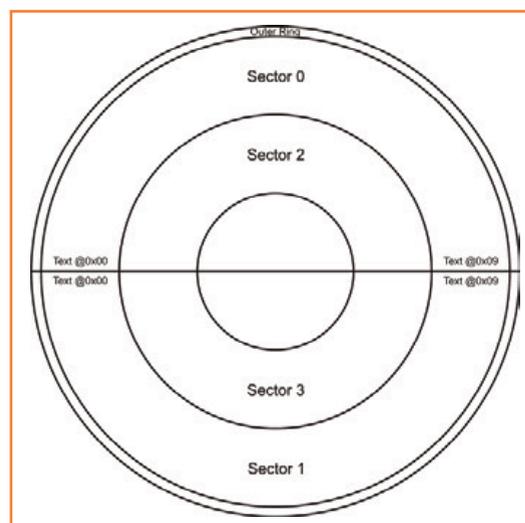
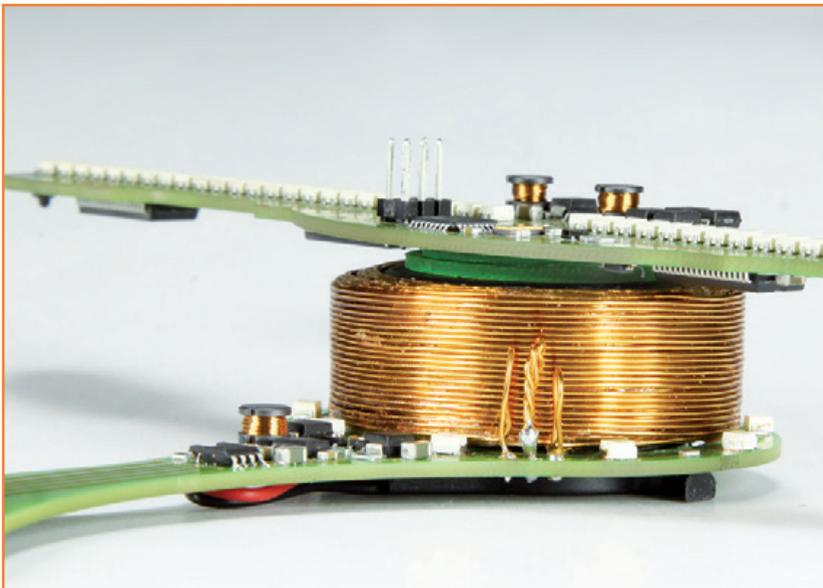


Bild 7. Display-Sektoren.

Die Echtzeituhr von Maxim DS3232 (U12) berechnet die Zeit und das Datum mit einem maximalen Fehler von 2 ppm (entsprechend 30 Sekunden pro Jahr) dank der thermischen Kompensation des integrierten Oszillators. Auch ohne Spannungsversorgung läuft die Uhr mehrere Tage weiter durch die Energie im Super-Kondensator C36. Wenn Sie die Uhr wieder einschalten, erscheint deshalb immer die aktuelle Zeit. Die Spannungsversorgung der Basiseinheit sorgt für zwei Spannungen. Für die Logik werden +5 V bereitgestellt (nur ein paar Milliampere). Eine stärkere Stromversorgung von +9 V ist für den Transformator, den Motor und die Infrarot-LEDs zuständig. Der Leistungsteil kann abgeschaltet werden, um die Energiereserven im Standby-Modus zu schonen.



## Fernbedienung

Ich habe die Uhr noch mit einem Infrarot-Fernbedienungsempfänger (U13) ausgestattet, der den Mikrocontroller über seine Input-Capture-Funktion anspricht. Dieser Receiver enthält einen Demodulator für 38-kHz-Signale bei einer IR-Wellenlänge von 950 nm. Sie können aber auch andere Typen einsetzen, je nach der Fernbedienung, die Sie verwenden möchten. Ich nutze die schöne weiße Fernbedienung eines Apple-MP3-Players [3]. Mit ihren nur sechs Tasten ist sie ideal für die Steuerung unserer Uhr. Sie versendet die Daten nach dem NEC-Protokoll, das eine logische Null durch einen 562  $\mu$ s langen 38-kHz-Burst und einer Pause derselben Länge darstellt. Eine logische Eins weist einen identischen Burst

auf, es folgt aber eine dreifach so lange Pause (also 1,62 ms). Jeder Druck auf eine der Tasten bewirkt, dass vier Bytes gesendet werden; die ersten beiden identifizieren das Empfangsgerät, die letzten beiden stehen für die gedrückte Taste.

## Software der Basiseinheit

Die *State machine* in **Bild 8** gibt eine Vorstellung vom Vorgehen der Software der Basiseinheit. Mit Counter 0 und Counter 2 werden die Signale für die induktive Energieübertragung erzeugt, während Counter 1 für die Decodierung der Fernsteuersequenzen zuständig ist. Um das Programm zu takten, habe ich das 1.024-kHz-Signal von U12 verwendet, womit ein Interrupt-Eingang angesteuert wird. Mit jeder Sekunde, die vergeht, wird die Echtzeituhr gelesen und der Wert an das Propellerdisplay übergeben. Das System bewertet die Umgebungshelligkeit (gemittelt über fünf Messungen) und stellt – wenn nötig – die Helligkeit des Displays nach. Der Drehencoder wird ebenfalls über einen Interrupt ausgelesen, nicht ohne softwaremäßige Tastenentprellung.

## Tempus fugit

Der Moment ist gekommen, auf die Pause-Taste zu drücken. Ich hoffe, ich habe Ihr Interesse geweckt und lade Sie ein, in der nächsten Elektor-Ausgabe alles über den mechanischen Aufbau der Propelleruhr zu erfahren. Um die Frustration aufgrund dieses *tempus interruptus* zu verringern, zeige ich Ihnen schon einmal ein Foto des Prototyps. Ich plane auch, eine Auswahl der Unterlagen über die Propelleruhr auf der Elektor-Website [1, 2] zur Verfügung zu stellen. Ich bin sicher, dass Sie die paar Wochen warten können, ich habe an diesem Projekt schließlich über sechs Jahre „herumperfektiert“!

(120732)

## Weblinks

- [1] 120732 – Propeller Clock – Montage.pdf
- [2] Downloadbare Software inklusive Sourcecode: [www.elektor-magazine.de/120732](http://www.elektor-magazine.de/120732)
- [3] Fernbedienung für Apple Universal Dock: <http://store.apple.com/de/product/MC746LL/A/apple-universal-dock?fnode=72>

# Propelleruhr (2)

## Mit ultra-stylishem Ziffernblatt

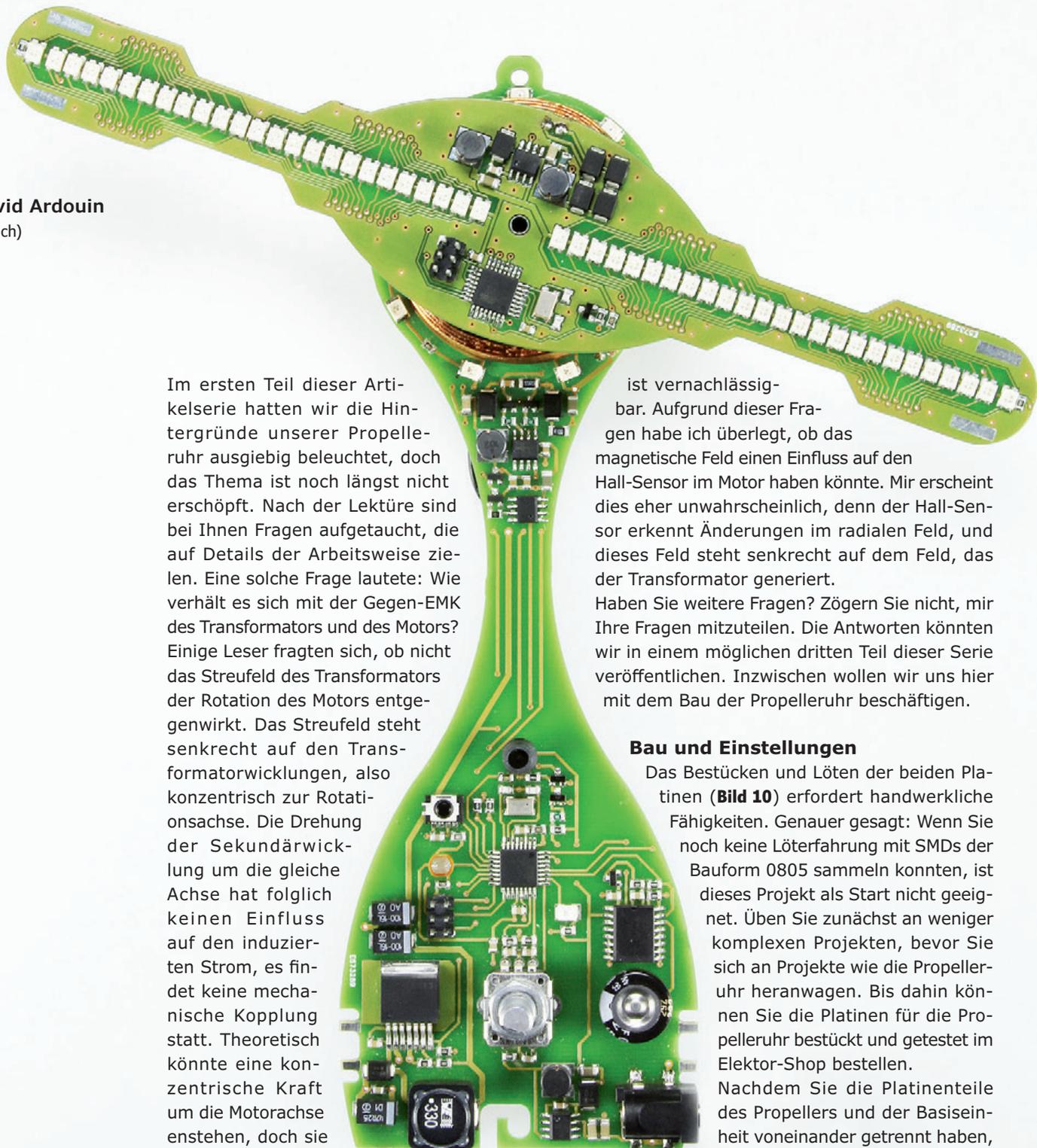
Von **David Ardouin**  
(Frankreich)

Im ersten Teil dieser Artikelserie hatten wir die Hintergründe unserer Propelleruhr ausgiebig beleuchtet, doch das Thema ist noch längst nicht erschöpft. Nach der Lektüre sind bei Ihnen Fragen aufgetaucht, die auf Details der Arbeitsweise zielen. Eine solche Frage lautete: Wie verhält es sich mit der Gegen-EMK des Transformators und des Motors? Einige Leser fragten sich, ob nicht das Streufeld des Transformators der Rotation des Motors entgegenwirkt. Das Streufeld steht senkrecht auf den Transformatorwicklungen, also konzentrisch zur Rotationsachse. Die Drehung der Sekundärwicklung um die gleiche Achse hat folglich keinen Einfluss auf den induzierten Strom, es findet keine mechanische Kopplung statt. Theoretisch könnte eine konzentrische Kraft um die Motorachse entstehen, doch sie

ist vernachlässigbar. Aufgrund dieser Fragen habe ich überlegt, ob das magnetische Feld einen Einfluss auf den Hall-Sensor im Motor haben könnte. Mir erscheint dies eher unwahrscheinlich, denn der Hall-Sensor erkennt Änderungen im radialen Feld, und dieses Feld steht senkrecht auf dem Feld, das der Transformator generiert. Haben Sie weitere Fragen? Zögern Sie nicht, mir Ihre Fragen mitzuteilen. Die Antworten könnten wir in einem möglichen dritten Teil dieser Serie veröffentlichen. Inzwischen wollen wir uns hier mit dem Bau der Propelleruhr beschäftigen.

### Bau und Einstellungen

Das Bestücken und Löten der beiden Platinen (**Bild 10**) erfordert handwerkliche Fähigkeiten. Genauer gesagt: Wenn Sie noch keine Lötferfahrung mit SMDs der Bauform 0805 sammeln konnten, ist dieses Projekt als Start nicht geeignet. Üben Sie zunächst an weniger komplexen Projekten, bevor Sie sich an Projekte wie die Propelleruhr heranwagen. Bis dahin können Sie die Platinen für die Propelleruhr bestücken und getestet im Elektor-Shop bestellen. Nachdem Sie die Platinenteile des Propellers und der Basiseinheit voneinander getrennt haben,



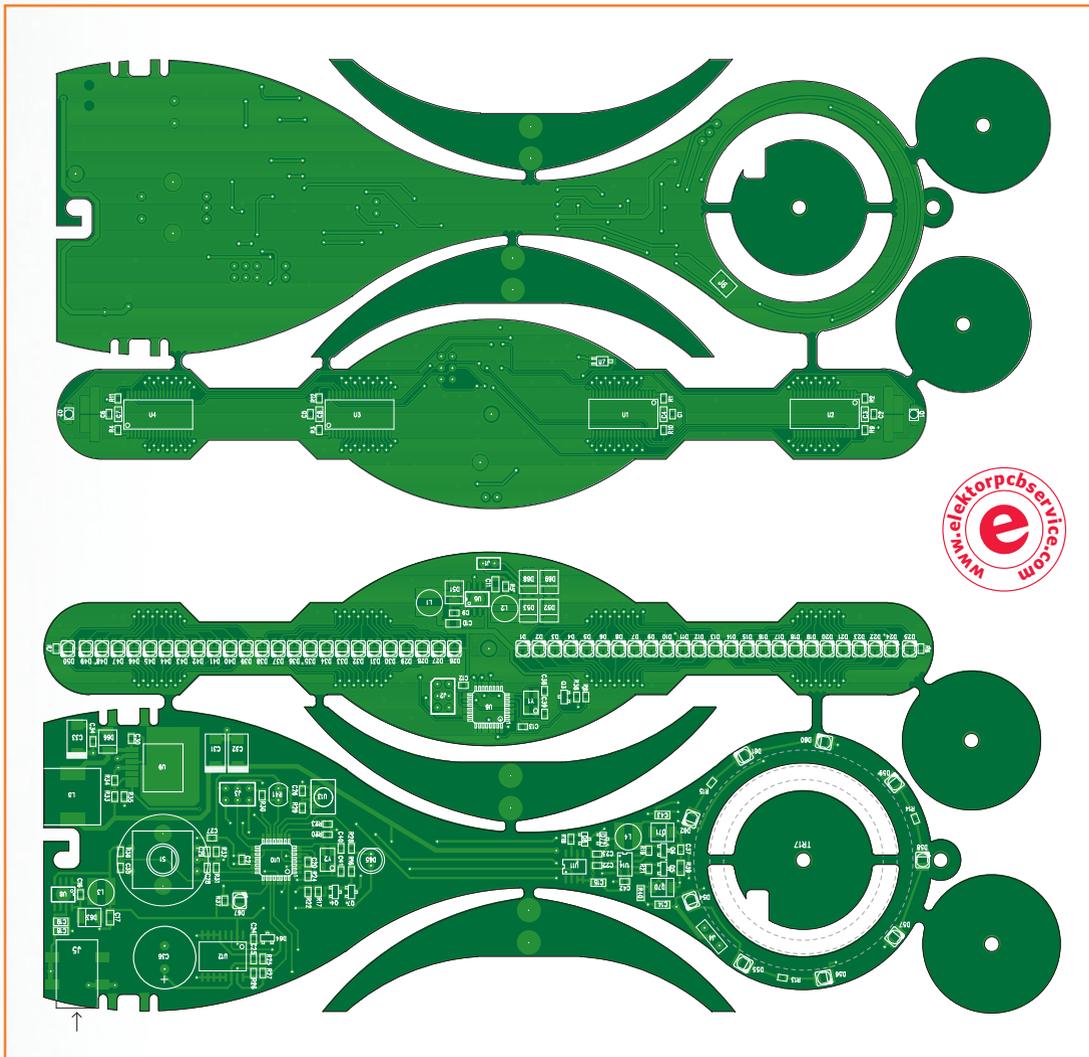


Bild 10a.  
Platine für die High-End-Propelleruhr. Meistens muss der Propeller nicht ausgewuchtet werden. Falls dies nötig sein sollte, können auf den rechteckigen Flächen an den Flügelenden kleine Mengen Lötlötn aufgetragen werden.

müssen Sie eventuelle Grate an den Platinenrändern mit einer flachen Feile beseitigen. Einige Teile haben lediglich mechanische Funktion, zum Beispiel die Zwischenringe (englisch: washer), die den Propeller bei der Montage auf Abstand vom Motor halten, sowie die halbmondförmigen Teile, die als Füße dienen.

Wenn es um das Löten von SMD-ICs geht, löte ich zuerst mit einem Lötkolben, der eine relativ breite Spitze hat. Fast alle IC-Anschlüsse werden beim Löten kurzgeschlossen, doch das ist unbedenklich. Wenn das IC montiert ist, entferne ich überschüssiges Lötmittel mit einem Stück Entlötlitze. Übrig bleibt dann nur das Lötmittel zwischen

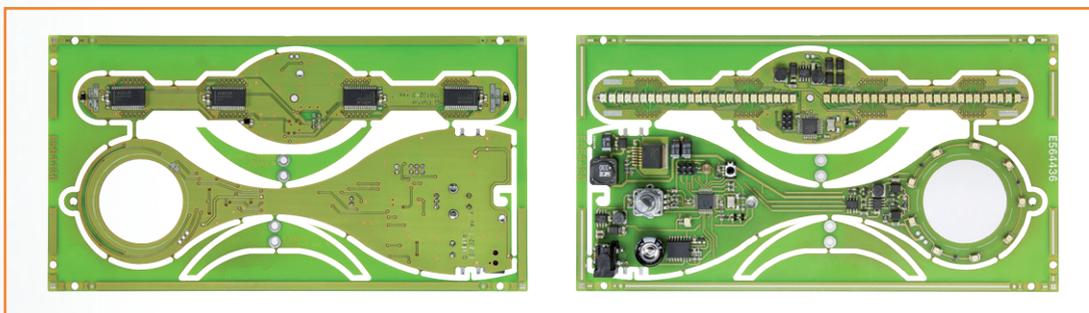


Bild 10b.  
Bei dieser Platinenversion fehlen noch die drei Zwischenringe (siehe Bild 10a und 13c).

Bild 11a.  
Fuse-Bits für den  
ATmega328 auf der  
Basiseinheit...

Fuses	Extended	0xFC	Brown-Out Trigger bei 4,3 V
	High	0xD9	Externer Takt 20 MHz, maximale Startzeit
	Low	0xFF	

Bild 11b.  
... und für den ATmega328  
auf dem Propeller.

Fuses	Extended	0xFD	Brown-Out Trigger bei 2,7 V
	High	0xD9	Externer Takt 20 MHz, maximale Startzeit
	Low	0xFF	

IC-Anschlüssen und Kupferbahnen. Bei den ICs U5 und U8 befindet sich eine Lötfläche auf der Unterseite. Zuerst löten Sie die Beine dieser ICs an die Platine. Anschließend löten Sie die IC-Fläche durch die dafür bestimmten Bohrungen von der Unterseite. Im PDF-Dokument zur Platine ist für diese Bohrungen der Durchmesser 0,5 mm angegeben. Bei den Platinen aus dem Elektor-Shop wurde der Durchmesser auf 3 mm geweitet, so dass die Spitzen üblicher LötKolben hindurch passen. Die Bohrung kann vollständig mit Lötmittel gefüllt werden, dies steigert die thermische Leitfähigkeit. Auf dem Propeller sind 50 SMD-LEDs zu montieren. Die LEDs müssen perfekt ausgerichtet sein, denn wenn sich der Propeller dreht, tritt die geringste Abweichung störend in Erscheinung. Die beiden LED-Reihen müssen perfekt symmetrisch zueinander stehen. Bereits im ersten Teil dieses Beitrags hatten wir den Grund beschrieben. Auf dem Foto unseres Prototyps ist erkennbar, dass die Infrarot-LED D65 eine Blende aus

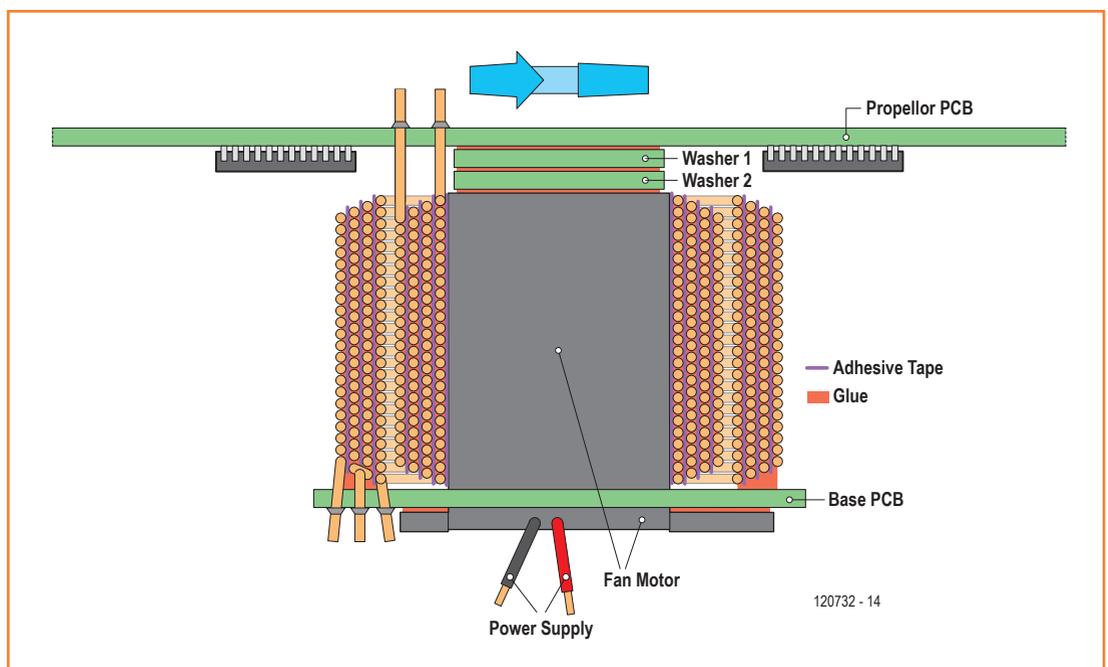
schwarzem Schrumpfschlauch erhalten hat. Die Blende sorgt dafür, dass das infrarote Strahlenbündel eng abgegrenzt wird. Das Signal, das beim Passieren der Propellerflügel entsteht, wird nur wenig verwaschen.

Die Software für die Mikrocontroller können Sie von der Projektseite [2] herunterladen und über die ISP-Schnittstellen in die Mikrocontroller übertragen. Für beide Mikrocontroller ist die Software fast identisch, sie unterscheidet sich jedoch in den Einstellungen der Fuse-Bits (siehe Tabellen in **Bild 11a** und **11b**). Die Unterspannungsschwelle (brown out) ist beim Mikrocontroller des Propellers niedriger eingestellt, sie wurde auf 2,7 V gelegt. Falls die Spannung des induktiv versorgten Propellers infolge einer plötzlichen Stromspitze einbricht, ist dem hiermit Rechnung getragen.

### Selbsttest

Bevor Sie den Bau der Propelluhr fortsetzen, ist ein Test beider Platinen angebracht. Testen Sie

Bild 12.  
Querschnitt durch den  
Drehmechanismus mit dem  
Transformator.



die Platinen zuerst einzeln, danach testen sie die Platinen gemeinsam. Der Propeller bleibt bei den Tests in Ruhestellung.

Beginnen Sie mit der Basiseinheit der Propelleruhr. Halten Sie S1 gedrückt, während Sie die Betriebsspannung einschalten. Die Test-LED D67 blinkt mit der Frequenz 2 Hz, was besagt, dass über die Infrarot-Verbindung Frames gesendet werden. Wenn Sie genau hinsehen, bemerken Sie bei LED D65 eine rötliche Färbung. Gleichzeitig muss der provisorisch angeschlossene Motor auf vollen Touren laufen, ein Zeichen dafür, dass die Spannung für den Transformator vorhanden ist. Mit zwei Pullup-Widerständen 1...10 k $\Omega$  zwischen den Pins 1 und 2 sowie 2 und 3 von J4 können Sie die steuernde Spannung für den Transformator auf einem Oszilloskop betrachten. Dort muss ein rechteckförmiges Signal mit der Frequenz 50 kHz und der Amplitude 9 V erscheinen.

Zweites Testobjekt ist der Propeller, zuerst allein. Schließen Sie eine 9-V-Batterie über J1 an, die Polarität ist beliebig. Die LEDs ganz nah an der Mitte müssen jetzt aufleuchten. Wenn Sie Q1 oder Q2 vor der Infrarot-Diode D65 bewegen, muss eine dritte LED aufleuchten. Das ist das Zeichen dafür, dass das Signal für die Propeller-Positionierung empfangen wird und der Propeller höchstwahrscheinlich funktionstüchtig ist. Anschließend halten Sie die Mitte des Propellers in Richtung der Basiseinheit, so wie er später auf den Motor montiert wird. Dabei ist U7 auf den Ring gerichtet, den die Infrarot-LEDs D54...D62 bilden. U7 befindet sich auf der Unterseite des Propellers. Die Propellerschaltung muss nun signalisieren, dass Daten-Frames empfangen werden. Das geschieht, indem die 50 LEDs auf den Propeller-Flügeln aufleuchten.

Haben Ihre Platinen diese Tests bestanden? Schön, dann können wir uns jetzt der mechanischen Konstruktion zuwenden.

## Mechanische Konstruktion

Die größte Tat in dieser Bauphase ist die Konstruktion des Transformators. Nehmen Sie einen PC-Lüfter mit 80 mm Durchmesser und entfernen sie außer dem Motor alle Teile einschließ-

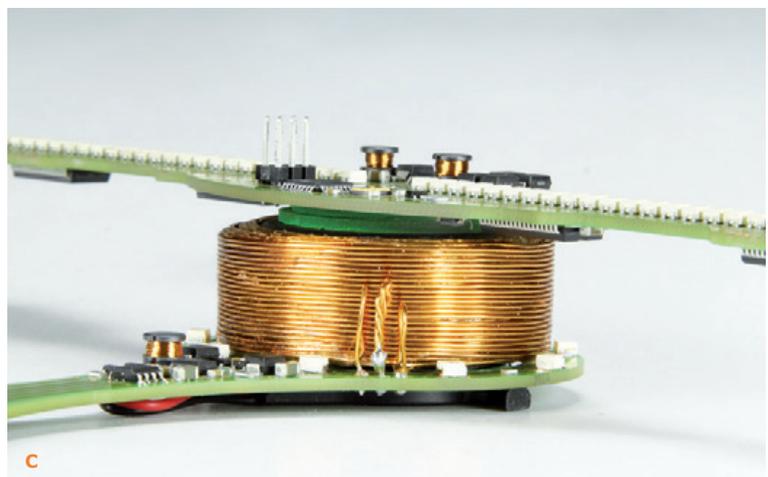
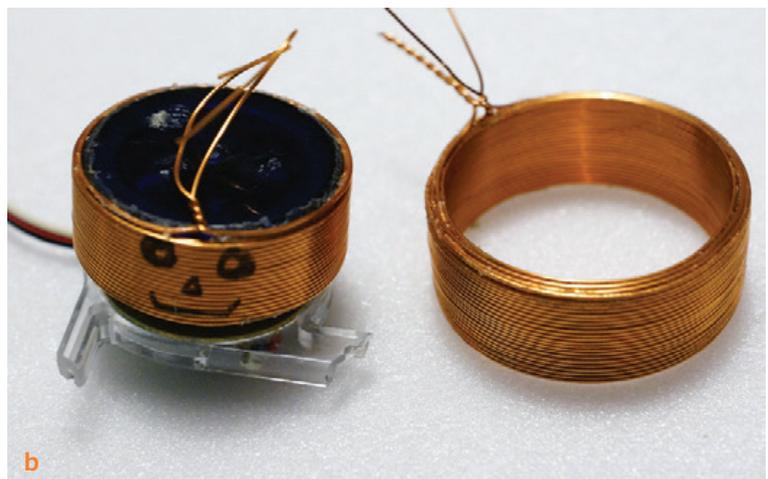
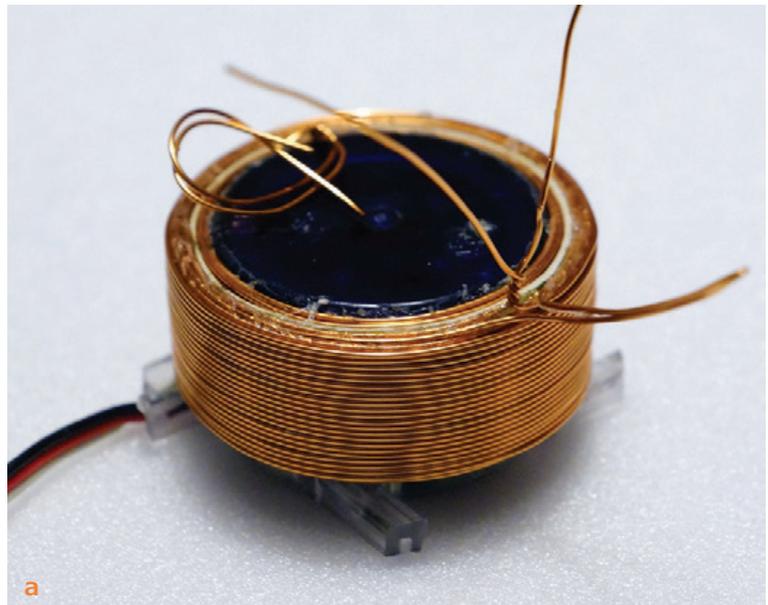


Bild 13.

Nahaufnahmen des Transformators: Nach dem Wickeln (a), einzeln (b) und montiert (c). Bei (a) ist der Pappring sichtbar, der die primäre und sekundäre Wicklung auf Abstand hält, und (c) zeigt die Zwischenringe, die den Propeller auf Abstand von der Basiseinheit halten.

lich Lüfterschaufeln. Dann wickeln Sie dort den Transformator auf. An dieser Stelle müssen wir nicht ins Detail gehen, denn der Arbeitsablauf wird schrittweise in einem besonderen Dokument erklärt, das Sie von der Projektseite [1] herunterladen können. Wenn Sie den Anweisungen strikt folgen und die nötige Geduld aufbringen, werden Sie den Transformator mit Erfolg wickeln. Die Schnittzeichnung in **Bild 12** und das Foto in **Bild 13** veranschaulichen, wie der Antrieb konstruiert ist. Die äußere Wicklung ist die Primärseite des Transformators, sie ist an der Basiseinheit fixiert. Auch der Stator des Motors wird an der Basiseinheit befestigt.

Die sekundäre Wicklung des Transformators dreht sich in der primären Wicklung, sie hat deshalb einen kleineren Durchmesser. Diese Wicklung wickeln Sie um den Rotor des Motors, den Propeller verbinden Sie später mit der Rotorachse. Wenn eine Lage gewickelt ist, legen Sie eine Schicht doppelseitiges Klebeband guter Qualität darüber, und darauf wickeln Sie die nächste Lage. Nehmen Sie Sekundenkleber, gering dosiert, um die Wicklungen zu fixieren.

Die beiden halbmondförmigen Teile der Platine sind die Füße, auf die Sie die Basiseinheit aufsetzen. Zuletzt löten Sie die Füße rechtwinklig an die Basisplatine. Falls Sie vorhaben, die Uhr an die Wand zu hängen, lassen Sie die Füße weg. Oben an der Basiseinheit befindet sich eine Öse, an der die Propelluhr aufgehängt werden kann. Der Haken unten in der Mitte ist als Zugentlastung für das Stromversorgungskabel gedacht. Natürlich darf das Kabel nicht mit dem Propeller

in Berührung kommen.

Wir möchten Sie daran erinnern, dass einige Bauteile mechanisch empfindlich sind. Dies gilt insbesondere für Q1 und Q2 an den Flügelspitzen. Achten Sie darauf, dass die Flügelspitzen nirgends anstoßen, wenn sich der Propeller mit hoher Geschwindigkeit dreht.

## Uhr mit Charakter

Nehmen Sie sich Zeit, kurz durchzuatmen! Wenn Ihre Platinen die Tests bestanden haben und alle mechanischen Arbeiten geschafft sind, müssen Sie nicht mehr lange warten, bis Sie Ihre High-End-Propelleruhr in vollem Glanz betrachten können. Was jetzt noch fehlt, ist ein Netzteil, das 10...14 V bei einem Strom von mindestens 250 mA liefert. Wenn Sie die Betriebsspannung einschalten, initialisieren die Mikrocontroller die Systeme, danach beginnt der Motor zu laufen. Zuerst leuchten die LEDs D1 und D26 auf zum Zeichen dafür, dass die Betriebsspannung vorhanden ist. Einige Sekunden später signalisiert LED D65, dass die Zeitanzeige startet. Jetzt muss das virtuelle Ziffernblatt die Zeit und darunter einen Willkommenstext anzeigen.

Nachdem Sie bis hierhin gekommen sind, können Sie die Propelleruhr mit der Fernbedienung (**Bild 14**) oder mit dem Drehknopf der Basiseinheit einstellen. Wenn Sie den Knopf drücken, versetzen Sie die Uhr in den Standby-Betrieb, oder Sie holen die Uhr von dort zurück. Drehen Sie nun den Knopf nach rechts oder links, dann durchlaufen Sie einige Dutzend Anzeigevarianten. Unterschiedliche Farbkombinationen, Ziffernblät-

## Stückliste

### Widerstände (SMD 0805, 5 %):

R1...R5, R17, R30 = 47 kΩ  
R6, R7, R16, R20, R22, R23, R24, R26, R27,  
R31, R32, R36, R37 = 4,7 kΩ  
R8...R12 = 22 kΩ  
R13...R15, R18, R19, R21, R29 = 100 Ω  
R25, R34 = 1,5 kΩ  
R28 = 330 Ω  
R33, R35 = 10 kΩ  
R38 = 1 MΩ

### Widerstände (SMD 1206, 5 %):

R39 = 10 Ω  
R40 = 0,1 Ω

### Kondensatoren, keramisch (SMD 0805, 20 %):

C1, C2, C5, C6, C12, C13, C20, C21, C22, C23,  
C24, C25, C27, C34, C42 = 100 n / 50 V

C9, C16, C30 = 10 n / 50 V  
C26, C28, C29, C35 = 1 μ / 10 V  
C37 = 1 n / 50 V  
C38...C41 = 33 p / 10 V

### Kondensatoren, keramisch (SMD 1206, 20 %):

C3, C4, C7, C8, C10, C17 = 22 μ / 10 V  
C11, C18, C19, C43 = 1 μ / 50 V  
C14, C15 = 10 μ / 16 V

### Kondensatoren, sonstige:

C31, C32 = 100 μ / 16 V, Vishay - 593D  
C33 = 47 μ / 25 V, Vishay - 593D  
C36 = 0,1 F / 5,5 V, Panasonic -  
EECF5R5U104 (9692703)

### Induktivitäten:

L1, L3 = 68 μH, 0,84 A, Bourns - SDR0604-

680KL (1828011)  
L2 = 220 μH, 0,38 A, Bourns - SDR0604-  
221KL (1828016)  
L4 = 1 mH, 0,12 A, Bourns - SDR0604-102KL  
(1828020)  
L5 = 33 μH, 3 A, Würth - 744771133  
(2082608)

### Diode:

D1...D50 = LED, zweifarbig PLCC4, Farben  
nach Wahl, z. B. Vishay VLMKE3400-GS08  
(1328370)  
D51...D53, D63, D66, D68, D69 = ON -  
MBRS140T3G (9557237)  
D54...D62 = Infrarot-LED, Osram - SFH421-Z  
(1226346)  
D64, D72 = BAT54  
D65 = VSLY5850 Infrarot-LED, Vishay

ter, Zeigerformen mit oder ohne Sekundenzeiger, die Datums- und Temperaturanzeige sowie die analoge oder digitale Darstellung stehen zur Wahl. Wenn Sie gefunden haben, was Ihrem Geschmack entspricht, ändern Sie während der nächsten zehn Sekunden nichts. Ihre Wahl wird dann permanent gespeichert, so dass die Propelleruhr beim nächsten Einschalten im gewählten Anzeigemodus läuft.

Halten Sie den Knopf zwei Sekunden gedrückt, dann gelangen Sie in das Konfigurationsmenü. Dort können Sie Zeit, Datum, Sprache, Rotationsgeschwindigkeit, die Tages- und Nachtanzeige und die Helligkeit einstellen. Dieses Menü durchlaufen Sie, indem Sie den Knopf kurz drücken. Den gewünschten Wert stellen Sie durch Drehen des Knopfs ein. Falls Sie für den Nachtbetrieb die Geschwindigkeit Null wählen, nimmt die Uhr ihren Betrieb erst auf, wenn Licht den Raum erhellt. Das dürfte ideal für das Schlafzimmer sein! Dann allerdings weigert sich die Uhr anzulaufen, wenn die Lichtintensität unter der Einschaltsschwelle liegt. Bevor Sie einen Defekt mutmaßen, bringen Sie die Propelleruhr in eine hellere Umgebung, anschließend können Sie die Einschaltsschwelle niedriger einstellen.

Damit sind wir am Ende des zweiten Teils dieser Serie angekommen. Ich hoffe, Sie haben beim Bau der Propelleruhr ebenso viel Freude, wie ich bei der Entwicklung und Realisierung hatte. Ich würde mich freuen, wenn Sie Ihre Erfahrungen und Ideen im Elektor-Forum mit anderen Lesern teilen!

(130389)gd

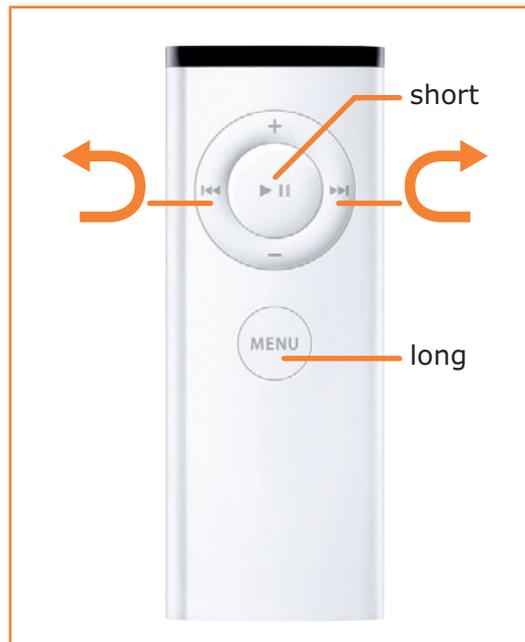


Bild 14.  
Steuerung mit einer Fernbedienung von Apple.

## Weblinks

- [1] Montageanleitung:  
[www.elektor-magazine.de/120732](http://www.elektor-magazine.de/120732)
- [2] Software-Download einschließlich Quellcode:  
[www.elektor-magazine.de/120732](http://www.elektor-magazine.de/120732)
- [3] Fernbedienung für Apple Universal Dock:  
<http://store.apple.com/de/product/MC746ZA/A/apple-universal-dock>

(1870807)  
D67 = PLCC2 LED orange  
D70, D71 = SMBJ48A TVS-Diode, Bourns  
(1899472)

### Halbleiter, sonstige:

Q1, Q2 = TEMT1020 Fototransistor, Vishay  
(1470165)  
Q3, Q4, Q7 = BC847A  
Q5, Q6, Q8 = IRLML0060 IR-Transistor  
(1783927)  
U1...U4 = MAX6957AAX Maxim [Digikey:  
MAX6957AAX+T-ND]  
U5, U8 = LM22674MR-5.0 (1679666)  
U6, U10 = ATmega328P-AU (1715486)  
U7 = SFH2400FA-Z Fotodiode, Osram  
(1226452)  
U9 = LM2670S-ADJ (1286849)

U12 = DS3231S, Maxim (1593292)  
U13 = TSOP6238 Infrarotsensor 38 kHz, Vishay  
(4913220)  
U14, U11 = TC4427ACOA, Microchip  
(1467705)

### Außerdem:

Y1, Y2 = Quarz 20 MHz, TXC -  
7B-20.000MAAJ-T (1841988)  
S1 = Incremental Encoder PEC11-  
4215F-S0024, Bourns (1653380)  
R41 = Fotowiderstand VT935G, Excelitas Tech  
(1652638)  
J2, J3 = Siftkontaktleiste, 2-3-polig, Raster  
2,54 mm  
J4 = Stiftkontaktleiste, 3-polig, Raster  
2,54 mm  
J5 = DC-Anschlussbuchse DC10AS 2,1 mm,

Cliff (1889309)  
J6 = PC-Lüfter 80 mm, 2000...3000 U/min  
TR1 = Transformator mit zwei konzentrischen  
Wicklungen [1]  
Lackierter Kupferdraht, Ø 0,56 mm, Länge  
etwa 25 m  
Sekundenkleber und dünnes, doppelseitiges  
Klebeband, siehe Montageanleitung [1]  
Platinen 120732-1...7

### Nachträgliche Korrekturen der Schaltung (Elektor Dezember 2013, S. 15/16):

- Bild 3a: R15 muss R12 heißen.
- Bild 3b: J1 ist keine Steckleiste, sondern der Anschluss von Trafo TR1 (siehe Bild 2).