The TSM-100
A Turntable Speed Measurement Device

Learn how to make your own microprocessor-controlled device that measures the speed of a turntable's platter. All the device's components are housed in a small transparent plastic box and an external power adapter is used for the power supply.

Photo 1: The completed TSM-100 device is shown encased in a transparent plastic box.

By
George Ntanavaras
(Greece)

The TSM-100 was developed with the Arduino Uno hardware board and the Arduino software IDE. The device indicates the result of the measurement on a LCD with a resolution of three decimal digits. The accuracy is guaranteed by using as internal time base a crystal oscillator accurate to ±50 PPM (±0.005%) with less than ±5 PPM aging per year. **Photo 1** shows the completed device.

How It Operates

Figure 1 shows a block diagram of the device. To begin, I used a compact photo micro sensor manufactured by Omron to sense the platter rotation. It incorporates an emitter of an optical beam and a detector as shown in Figure 1. When an object passes in front of its sensing area, it reflects the optical beam of the emitter, thus changing the amount of optical energy reaching the detector. The sensor incorporates a visible light-intercepting filter, which allows objects to be sensed without being greatly influenced by the light radiated from fluorescent lamps.

For the detection of the platter rotation, a small reflective area is attached to the side of the platter, and as it spins, it reflects back the light emitted by the LED every time the reflective area is in front of the sensor. The photo transistor is triggered, turns on, and produces a short duration pulse, which is driven to a Schmitt trigger circuit to increase the noise immunity of the input signal and remove other disturbances.

The digital output of this circuit is directly driven to the microprocessor for processing. It also triggers another circuit, which outputs a low level (0 V) signal indicating to the microprocessor that a valid input was detected. Additionally it triggers a circuit that flashes a green LED, indicating to the user that a valid pulse has triggered the device, making the set up for the operation of the unit easier.

There is a large variety of platters manufactured with different shapes and from different materials. **Photo 2** shows two of the most common types of platter and how I set up the device to perform the measurement. Photo 2a shows a turntable made

SAGA, $349

MAGNI 3, $99

WE CAN'T BEAT DIY.

We get it. No matter what we do, we can't deliver the amazing satisfaction of designing and building something that's uniquely your own. There's really nothing like it.

But...if you ever want high-value audio products—like a discrete current-feedback headphone amp for $99 or a microprocessor-controlled, relay-switched stepped attenuator preamp (with a push-pull hybrid tube buffer) for $349, then hey, well, here we are.

SCHIIT

WWW.SCHIIT.COM

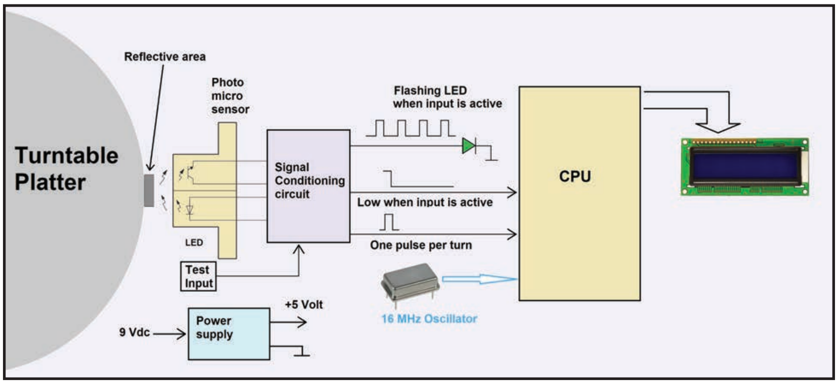

Figure 1: The block diagram of the device

from non-reflective material. In this example, I attached a 10 mm × 15 mm piece of self-adhesive highly reflective aluminum foil tape directly to the side of the turntable's platter. Then I placed the device with the photo sensor at a distance of about 5 mm to 6 mm and perpendicular to the reflective area.

Photo 2b shows a turntable's platter that is made from a more reflective material similar to aluminum, so I attached a small piece of self-adhesive black foam with dimensions of about 10 mm × 30 mm and a thickness of 10 mm to the side of the platter. I then stuck a piece of the highly reflective aluminum foil tape to the top of this foam and placed the device 5 mm from the reflective piece. With this setup, the device's sensor was 15 mm from the platter and could not be triggered until the reflective aluminum foil tape was in front of the sensor as the platter was turning around.

It is very important, in both cases, that the device remains stationary without any movement during the measurement. The rest of the processing is performed by the software running on the CPU, which then drives the LCD module to show the result of the measurement.

I wanted to increase the device's mobility by using a 9 V battery for the supply, but the consumption was a little high and the battery life would be relatively small. So I chose an external power adapter with an output voltage of 9 VDC.

The Electronic Diagram

Figure 2 shows the completed electronic diagram for the device. The operation is as follows. The 9 VDC of the power adapter output are connected through the connector J1 to the circuit. A slow-blow fuse, mounted on the PCB, protects the external AC/DC power supply from any short circuit that could happen within the device. Diode D1 protects the circuit from an accidentally reverse polarity and capacitors C1 and C2 filter the voltage. The IC3 regulator produce a stable output voltage of +5 VDC, while capacitors C3 and C9 improve the stability of the regulator. A small heatsink for TO-220 devices is used on the IC3 regulator with a thermal resistance of 25.9 °C/W and dimensions 9.52 mm × 13.21 mm × 19.05 mm.

The four pins of the photo sensor are connected as follows: the LED emitter is connected to the pins ANODE and CATHODE and the photo transistor is connected to the pins COLLECTOR and EMITTER. The LED current is set by the parallel combination of the resistors R7 and R8 at about 23 mA. The output of the photo detector drives resistor R10 and goes to the JP2 connector, which is used to select the input between the output of the photo detector and the external signal, which is connected to pin TEST_IN and can be used to check the device's accuracy. The selected input goes to the signal conditioning circuit comprising of the components C6, R3, R4, R5, D2, D3, and IC2F. Capacitor C6 should have a large value to pass correctly the pulse from the sensor and due to its DC biasing should be a bipolar type. After the removal of any disturbances by the signal conditioning circuit, a clear digital pulse is produced each time the detector passes in front of the small

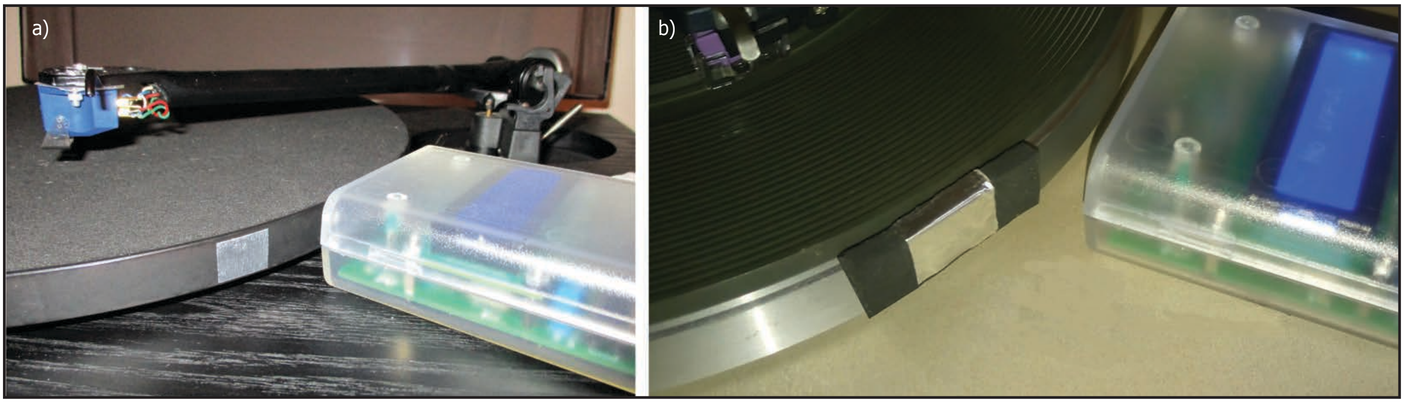

Photo 2a: This example shows the device being used with a platter made from a non-reflective material. b: I had to make a slight alteration because this platter is made from more reflective material.

reflective area, which is placed on the platter of the turntable. This pulse is inverted by IC2B and goes directly to the microprocessor Digital input pin 2.

The components IC2C, D4, R14, R6, C8, and IC2D produce a low pulse with a very long duration of about 2.4 seconds each time that the circuit is triggered by a positive output pulse by the photo detector. This signal goes to the digital input pin 7 of the microprocessor to indicate that a valid pulse has been detected. If this input pin is high, the microprocessor understands that no valid measurement exists and indicates on the display the message "No input."

In addition, the circuit around components D5, R12, C12, IC2A, and R16 oscillates with a frequency of about 6 Hz and shows to the user by blinking the green LED that the device is triggered correctly.

I used an external 16 MHz HCMOS clock oscillator (QG1) with 50 ppm accuracy to drive the CPU in order to have an accurate and stable time base for the measurement.

The microprocessor unit is the ATMEL ATMEGA328-PU. It is a low power CMOS 8 bit microcontroller based on the AVR enhanced RISC architecture.

All the processing of the measurement is performed by the software running in the CPU. The result of the measurement is shown to the Alphanumeric LCD module, which is connected to the JP1 connector. I used a small LCD module with two lines of 16 characters and dimensions 65.5 mm × 36.7 mm × 13.5 mm. The characters are white on a blue background and white LED back-light. Trimmer R13 adjusts the contrast of the LCD.

A tactile switch S1 is also included on the device to reset it, if necessary. A pull-up resistor R1 keeps the reset input of the CPU high. Connectors IOH, AD, and IOL are not used in this device.

Assembling the Device

I designed the PCB using the Eagle Layout editor Demo version, which can be downloaded for free. It is a fully operational version, with the only limitation being the maximum dimensions of the PCB, which was not a problem on this project.

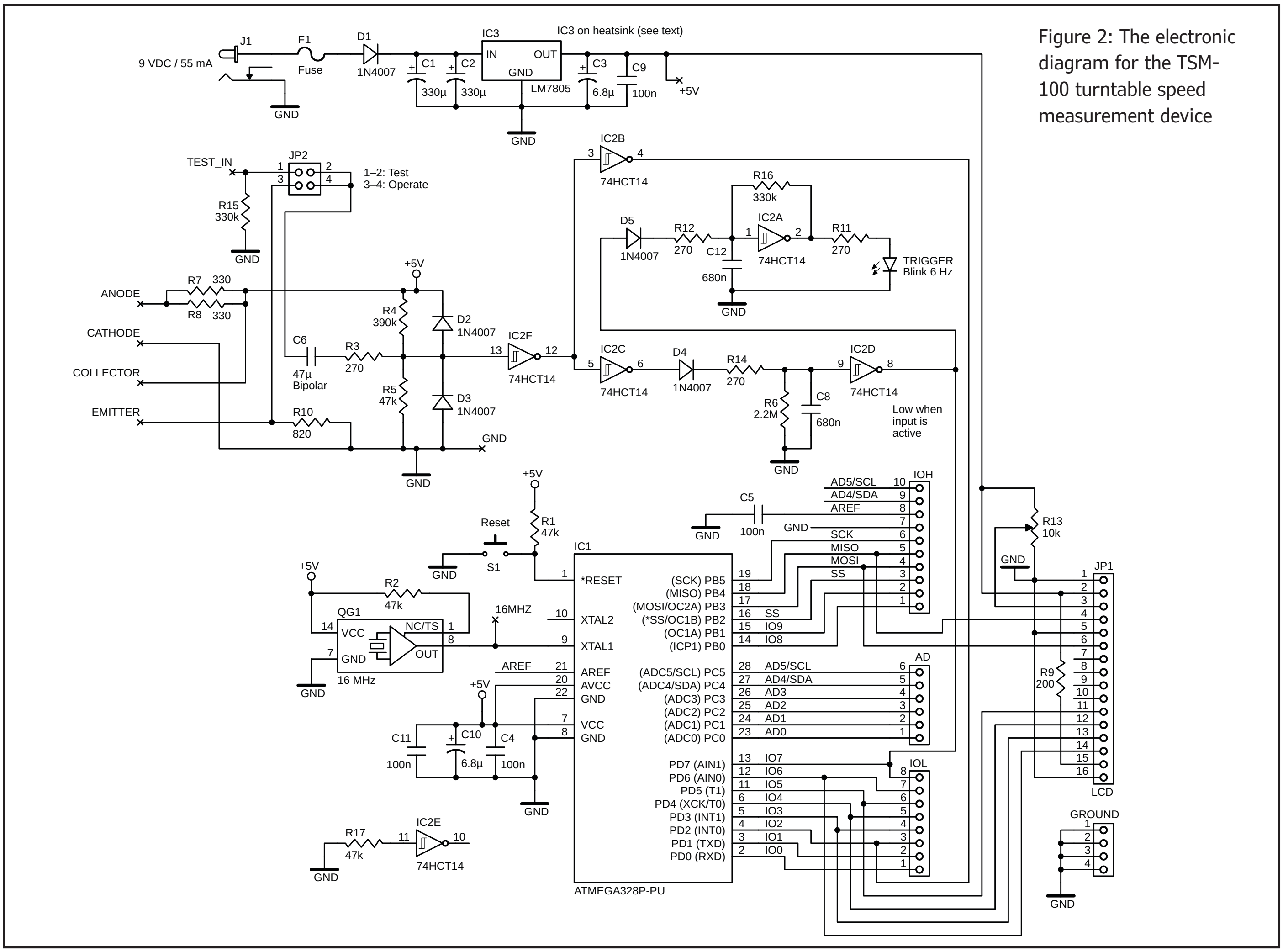

Figure 2: The electronic diagram for the TSM-100 turntable speed measurement device

I ordered a high-quality PCB manufactured from FR4 composite material with a 1.6 mm thickness and 35 µm copper solder on both sides and silkscreen on the top side.

The parts list for the TSM-100 project can be found in the Supplementary Material section of the audioXpress website (see Project Files at the end of the article). The guide for the assembly of the components is detailed in **Figure 3**. All the components are placed on the PCB, to simplify the construction. **Photo 3** shows some important construction details. When I performed the tests, I found out that the regulator IC3 (7805) was getting a little hotter than I would prefer so I decided to put it on a small heatsink for better cooling. To ensure a safe distance from diode D1, which is very close, I put a small gap between the heatsink and the PCB by inserting an additional nut as shown in Photo 3.1. The heatsink is then placed on the top of this nut and another M3 nut secures the regulator, the heatsink and the PCB together as shown in Photo 3.2. The distance from the PCB offers also better cooling for the heat sink and lowers the temperature.

The photo sensor is placed on the PCB as shown in Photo 3.3. The procedure is as follows: First the four terminals of the sensor are bent and inserted in the holes of the PCB, then a M2x12 screw with a nut secures the sensor on the PCB, and finally the four pins are soldered.

I used a small semi-transparent plastic enclosure with dimensions 124 mm × 72 mm × 30 mm to house the PCB, so that the LCD could be read easily without any opening to the top cover of the plastic box. At the low side of the plastic box, I opened a 12 mm hole so that the connector of the external power adapter could easily enter to the device without opening the plastic box. This opening is also necessary to provide some ventilation for the internal heat that is developed in the device. The PCB is screwed in four points to the bottom of the box. Details are shown in Photo 3.4 and Photo 3.5.

I used nylon washers between the PCB of the LCD and the nuts of the stand-off that hold the LCD as shown in Photo 3.6. This insulator is necessary because otherwise the tightening of the nut may destroy the PCB tracks of the LCD that are very close.

When the device was assembled, I powered it on and measured all the supply voltages to verify that they were right according to the electronic diagram. On the first power on, the LCD is blank and with the trimmer R13, the contrast should be adjusted, until the alphanumeric characters are easily read. The jumper JP2 should be placed on the position 1-2 (Test) for the testing of the device or to the position 3-4 (Operate) for the normal operation. For the testing an accurate low frequency square wave of around 0.5 up to 1 Hz at a level of 5 V should be connected to the TEST_IN and GND pin. The frequency of the signal multiplied by 60 will be indicated on the LCD. For example, connecting an accurate 0.5 Hz square wave gives 0.5 × 60 =

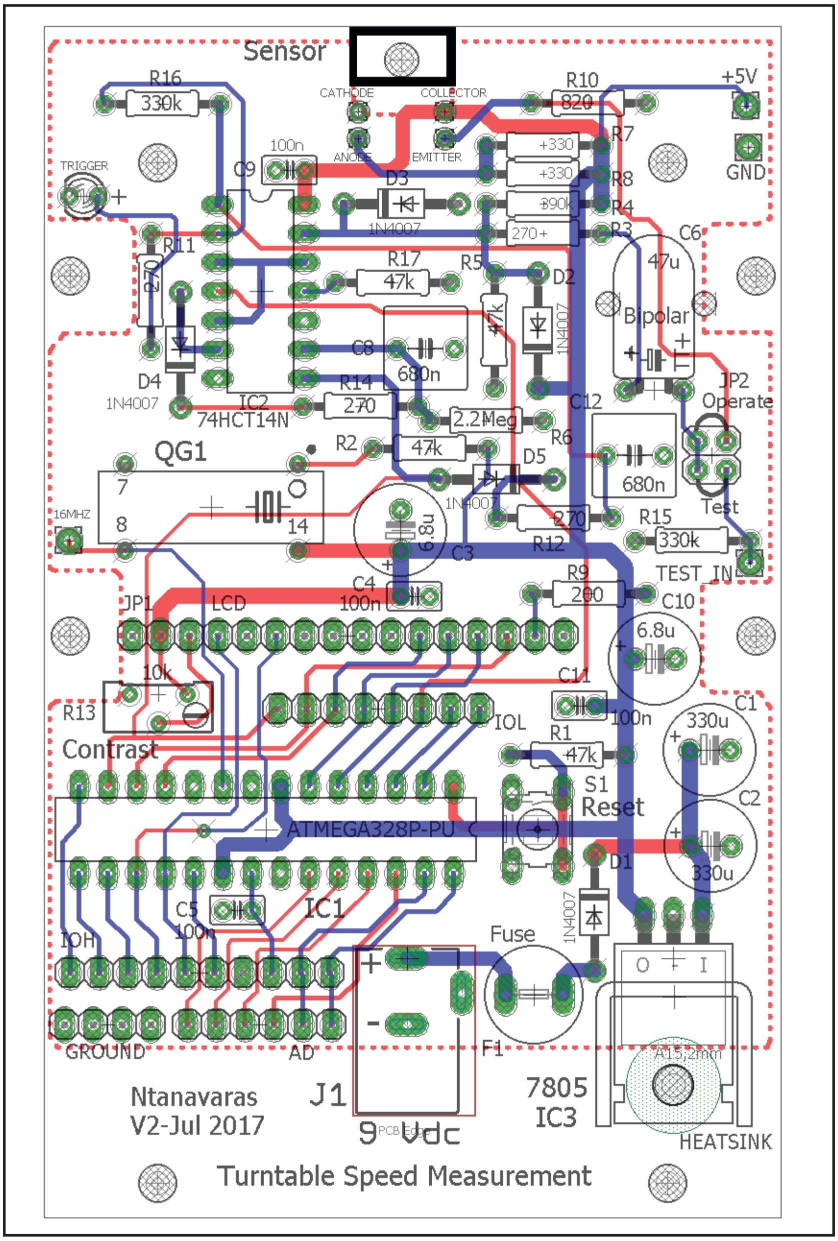

Figure 3: This is the PCB assembly of TSM-100 device.

About the Author

George Ntanavaras graduated from the National Technical University, Athens, Greece, in 1986 with a degree in Electronic Engineering. He currently works in the Development Department for a Greek electronics company. He is interested in the design of preamplifiers, active crossovers, power amplifiers, and most loudspeakers. He also enjoys listening to classical music.

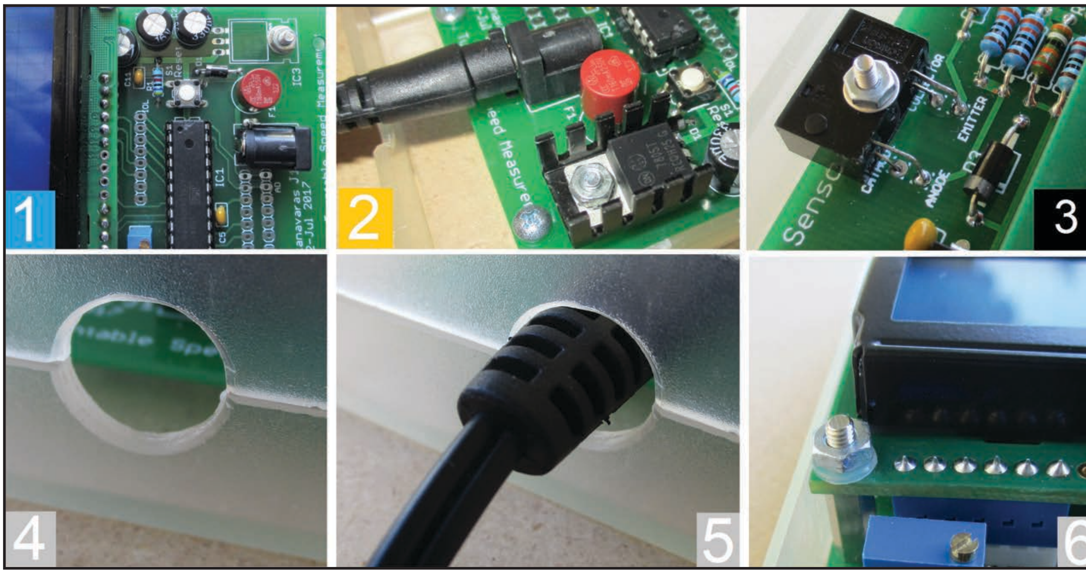

Photo 3: Important construction details are depicted.

30 RPM, and the display will indicate this with three decimal digits as 30.000 RPM.

The Device's Software

I used the Arduino Uno REV 3 hardware board during the development phase of the project. This is a microcontroller board based on the ATMEL ATmega328P. It has 14 digital input/output pins, six analog inputs, a 16 MHz quartz crystal, a USB connector to a PC, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller. It is connected to the computer with an USB cable, which also will power the board or otherwise can be powered with an AC-to-DC adapter or even a battery.

For the software development, I used the Arduino Integrated Development Environment (or Arduino Software IDE). This supports a text editor for writing the code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It is an easy tool, aimed at those without a background in programming.

I wrote a simple software code for the operation of the device. It calculates the "rpm" for each revolution of the platter and the result is shown to the LCD with a resolution of three decimal digits.

Figure 4 shows how it operates. Each time the reflective area is in front of the sensor, a short pulse is produced. The rising edge of this pulse triggers a CPU interrupt to run a short subroutine that uses the Arduino "micros ()" function, which returns the number of microseconds since the CPU start running the program. This function is very convenient but it has some shortcomings. The returning number will overflow after approximately 70 minutes of operation going to zero value and has a resolution of 4 μs since the returned value is always a multiple of 4. Both of them are not serious problems for this application because the measurement of the "RPM" is performed in every turn of the platter and the result is shown with only three decimal digits.

The calculated time difference T between two successive readings is the period for one revolution and for the most common speeds of 33 1/3 and 45 RPM, the period T is 1.8 s and 1.33 s, respectively. These time periods are very long and

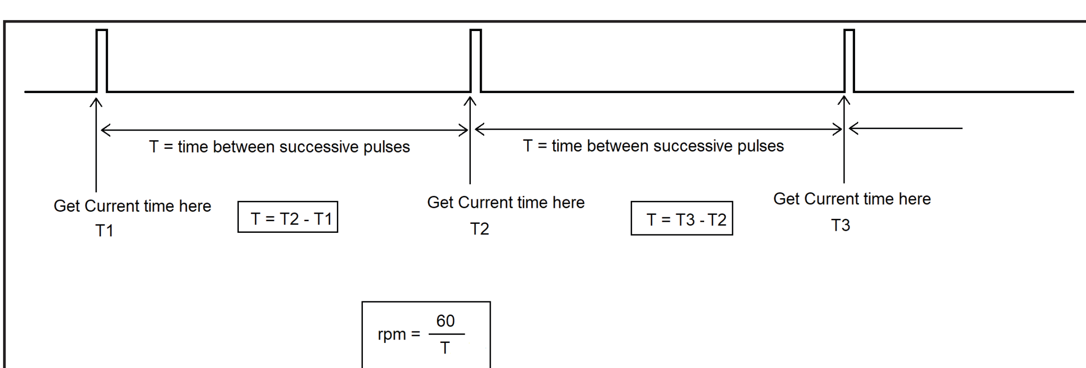

Figure 4: Operation of the software

```
//
//                        Turntable Speed Meter ......V1.0
//                              by George Ntanavaras
//
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 6);    // LCD pin connections
volatile unsigned long Cur_Time;          // Arduino Current time in micro seconds
volatile long Ttime;                      // Storing total time of a cycle
volatile unsigned long Ttime1;            // Storing start time of a cycle
volatile unsigned long Ttime2;            // Storing stop time of a cycle
volatile int meas ;                       // number of succesive measurements
volatile float frequency;                 // Storing frequency
volatile float rpm;                       // Storing rpm
volatile boolean trigger;
volatile boolean in_active ;
//-----------------------------------------------------------------------------
void isr ()
{
   Cur_Time = micros();
   meas=meas+1 ;
   if (trigger) Ttime1 = Cur_Time ;       // get end time
   if (!trigger) Ttime2 = Cur_Time ;      // get start timereturn ;
   trigger = !trigger ;
}
//-----------------------------------------------------------------------------
void setup()
{
   trigger = false ;
   Ttime1 = 0 ;
   Ttime2 = 0 ;
   init();
   meas =0 ;
   pinMode(2, INPUT);
   pinMode(7, INPUT);
   attachInterrupt(0, isr, FALLING);
   lcd.begin(16, 2);            // set LCD = 16 char by 2 lines
   lcd.clear();                 // clear the whole LCD
   lcd.display();
   delay(1200);
   lcd.clear();                 // clear the whole LCD
   lcd.setCursor(0,0);          // set cursor to the first character, top line
   lcd.print("TSM-100 V1.0");   // label the LCD top line
   delay(1200);
   detachInterrupt(0);
   if (trigger) Ttime1 = Cur_Time ;   // get end time
   if (!trigger) Ttime2 = Cur_Time ;  // get start time
   attachInterrupt(0, isr, FALLING);
   lcd.clear();                 // clear the whole LCD
   lcd.setCursor(0,0);          // set cursor to the first character, top line
   lcd.print("Turntable Speed"); // label the LCD top line
   lcd.setCursor(0,1);          // set cursor to the first character, second line
   lcd.print(" Meter");         // label the LCD top line
   delay(1200);
   detachInterrupt(0);
   if (trigger) Ttime1 = Cur_Time ;   // get end time
   if (!trigger) Ttime2 = Cur_Time ;  // get start time
   attachInterrupt(0, isr, FALLING);
   lcd.clear();                 // clear the whole LCD
   lcd.setCursor(0,0);          // set cursor to the first character, top line
   lcd.print("by");
   lcd.setCursor(0,1);          // set cursor to the first character, second line
   lcd.print("G. Ntanavaras");
   delay(1000);
   detachInterrupt(0);                                               (continued)
```

Figure 5: Here is a listing of the software code that I wrote.

Figure 5: continued

```
    if (trigger) Ttime1 = Cur_Time ;   // get end time
    if (!trigger) Ttime2 = Cur_Time ;  // get start time
    attachInterrupt(0, isr, FALLING);
    lcd.clear();                  // clear the whole LCD
    lcd.print("Wait ");
    lcd.setCursor(0,1);           // set cursor to the first character, second line
    lcd.print(" to measure ...");
    delay(2000);
    }
//-----------------------------------------------------------------------
void loop ()
{
in_active = digitalRead(7);       // Check if there is a valid input to measure
if (in_active == LOW)
   {
   if (meas < 2 ) return ;        // If there is not a new measurement, return
                                          //
                                          // New measurement
                                          //
   Ttime = Ttime1-Ttime2 ;                // Calculate period in microseconds
   Ttime = abs (Ttime) ;
   frequency=Ttime+88 ;                   // Compensate to improve accuracy
   frequency=1000000 / frequency;         // calculate frequency from Ttime
   rpm=60*frequency;                      // calculate RPM from frequency
   // Display the readings
   lcd.clear();                           // Clear the whole LCD
   lcd.setCursor(0,0);          // Set cursor to the first character, first line
   lcd.print (rpm, 3);
   lcd.print (" RPM");
   meas=1;   // Prepare for the next measurement
}
else
    {
    lcd.clear();                          // clear the whole LCD
    lcd.print (" No input ");
    delay(100);
    }
}
// end of loop and program
//--------------------------------
//--------------------------------
```

the microprocessor unit can easily measure them with high accuracy.

The frequency is calculated as the inverse of the period: frequency = 1/T.

The revolutions per minute (RPM) of the turntable's platter are calculating them as RPM = 60 × frequency.

The selection of the software commands within the interrupt routine was very critical and after a lot of experimentation, I managed to have a measurement with high stability and repeatability. A small constant compensation (equal to 88) was added to the computed time for better accuracy.

The software has excellent measurement accuracy for frequencies up to 1.4 Hz (84 RPM).

The software should be downloaded to the ATmega328P CPU using the Arduino Uno board since the TSM100 device does not have such capability. After the programming, the CPU is removed from the Arduino Uno board and placed on the IC1 socket of the PCB of the device.

A complete list of the software code is shown in **Figure 5**. It has the typical structure of an Arduino program, which is called the "sketch." It starts with the declaration of the variables that are used by the program. The short interrupt routine "void isr ()" then follows with the "micros ()" commands. The main program starts with the "setup ()" which is executed only once when the program starts to initialize the parameters and continues with the "loop ()" which runs continuously, computes the RPM, and indicates the result to the LCD module (see **Photo 4**).

The Reference Oscillator

The accuracy of the measurement depends strongly on the software and during the development phase I was looking for an easy way to check the various versions of the software and adjust them to perform as accurately as possible. For this reason, I

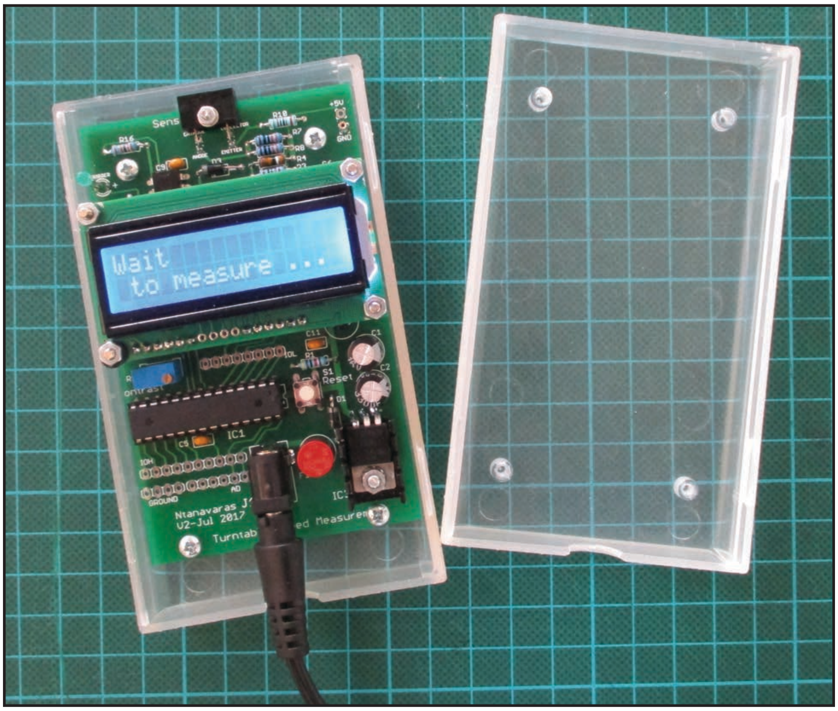

Photo 4: This is an internal view of the TSM-100 device.

designed a special unit with an accurate and stable clock output which I called it "Reference Oscillator."

The unit is based on an Oven-Controlled Crystal Oscillator manufactured by Abracon (AOCJYR-10.000MHZ-M5625LF). It is an expensive SMD component, with a ±0.5 ppm initial frequency tolerance at 25°C and a stability as low as ±25 ppb (note that it is parts per billion, not parts per million) over the temperature range from -40 to +85°C.

Figure 6 shows the completed electronic diagram. The output of the 10.000 MHz Oven Controlled Crystal Oscillator is at a level of 3.3 V and the MC74VHC1GT04 CMOS buffers and shifts this level to 5 V so that the rest of the circuit can operate smoothly. The 10 MHz clock is driven to eight stages of dividers. The seven of them divide by 10 while the last one divides by 2. The output of each divider is connected to a 12-position switch, which selects the signal that will be sent to the output connector from one of the following frequencies: 10 MHz, 1 MHz, 100 kHz, 10 kHz, 1 kHz, 100 Hz, 10 Hz, 1 Hz, and 0.5 Hz.

The selected signal is buffered by another MC74VHC1GT04 buffer inverter before it is connected to the output connector.

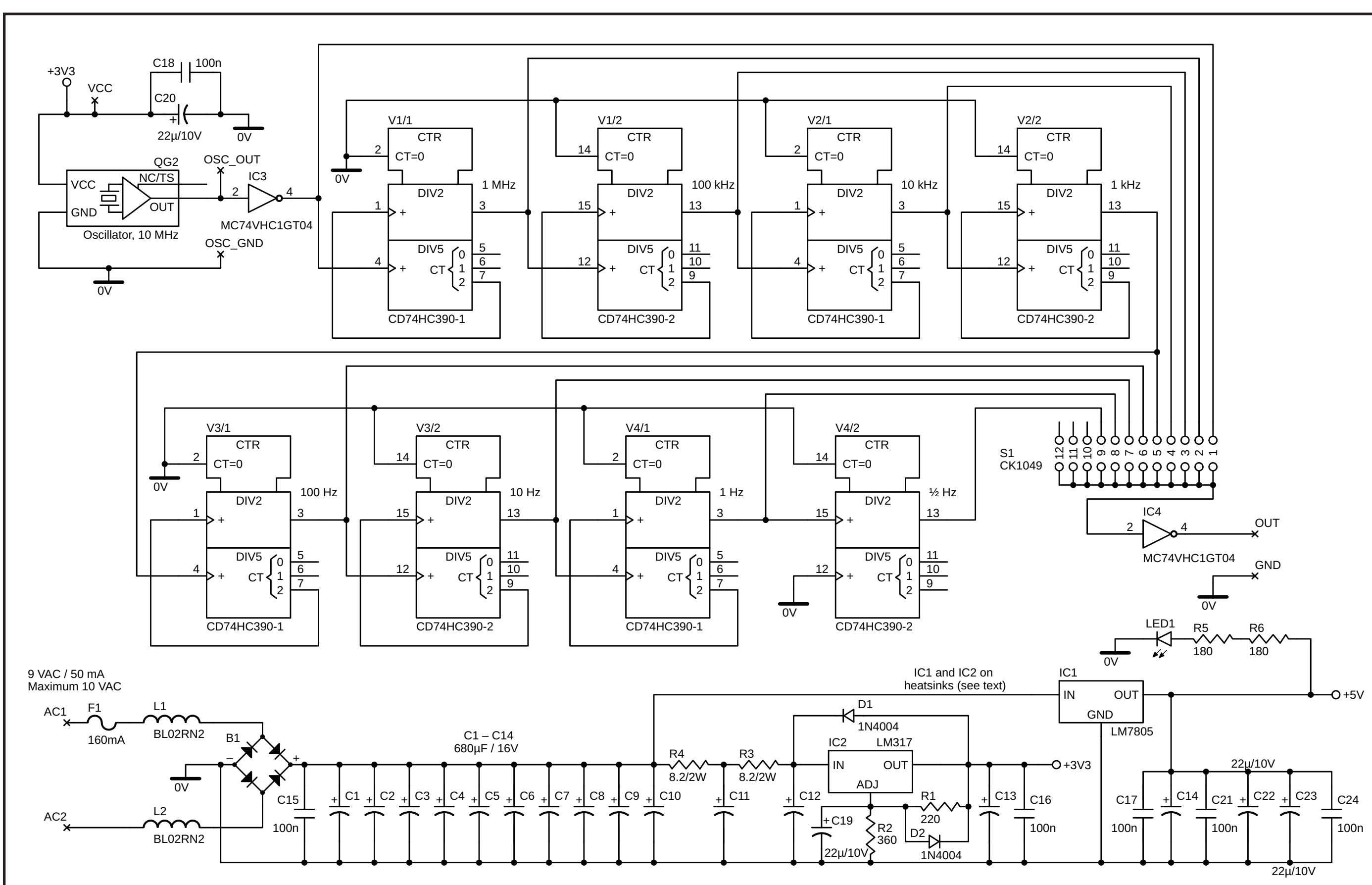

Figure 6: The electronic diagram of the Reference Oscillator

For the supply of the oscillator, I used a Class B, external plug-in AC/AC power supply. Its mains input is protected with a 130°C thermal fuse and has a 9 VAC/10 VA secondary output which through the front panel connector goes to pins AC1 and AC2 of the circuit.

A slow-blow fuse, mounted on the PCB, protects the external AC/AC power supply from any short circuits that could happen within the device. L1 and L2 are ferrite beads inductors that have small impedance at low frequencies, which increases to a high impedance at high frequencies. They offer electromagnetic induction (EMI) protection from several megahertz up to gigahertz by suppressing any noise coming from the power supply.

The AC voltage is rectified by the bridge B1 and filtered by the capacitors C1 to C10. Resistors R3, R4 and capacitors C11, C12 form low-pass filters to smooth the ripple of the input voltage before it enters to the 3.3 VDC regulator. Two voltage regulators are used, a 7805 for the 5 VDC and a LM317 for the 3.3 VDC. The regulators IC1 and IC2 required some cooling and I used two small heatsinks to keep their temperature low.

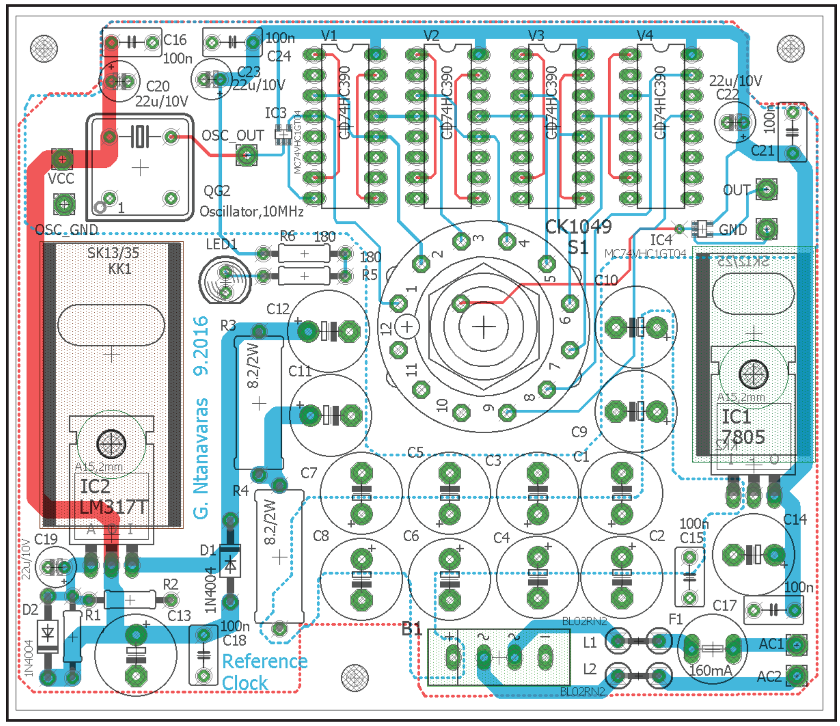

Figure 7: Here is the PCB assembly of the Reference Oscillator.

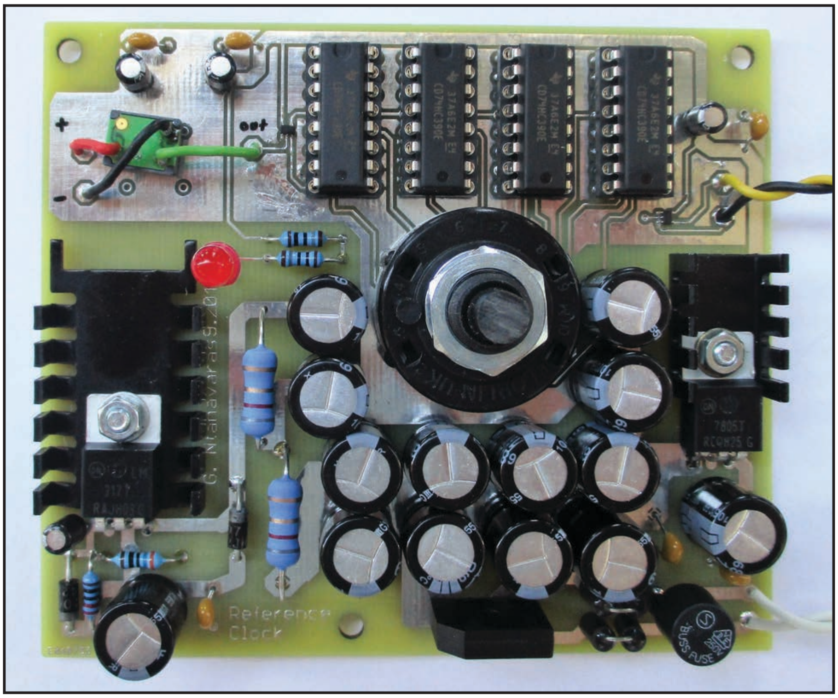

Photo 5: This is a top view of the Reference Oscillator's completed PCB.

Additional supply filtering is offered by capacitors at the output of the regulators and near the Oven-Controlled Crystal Oscillator and the other ICs. An LED illuminates to indicate the operation of the circuit.

I designed another PCB for the oscillator using the Eagle Layout editor Demo version and I ordered a prototype board without any silk screen to assembly the circuit.

The parts list for the Reference Oscillator can be found in the Supplementary Material section of the audioXpress website (see Project Files at the end of the article). The guide assembly is detailed in **Figure 7**, while **Photo 5** shows the final assembled PCB.

The Oven-Controlled Crystal Oscillator that I finally used was different from the initial one I had selected when designing the PCB, and for this reason, the new oscillator was connected to the PCB with short lengths of wire as shown in the left top corner of Photo 5. I used two small heatsinks for each regulator. For the LM317T regulator, I used a heatsink with a temperature constant of 21°C/W and with the dimensions 6.7 mm × 22 mm × 35.6 mm. For the 7805 regulator, I used a heatsink with a temperature constant of 25.9 °C/W and with the dimensions 9.52 mm × 13.21 mm × 19.05 mm.

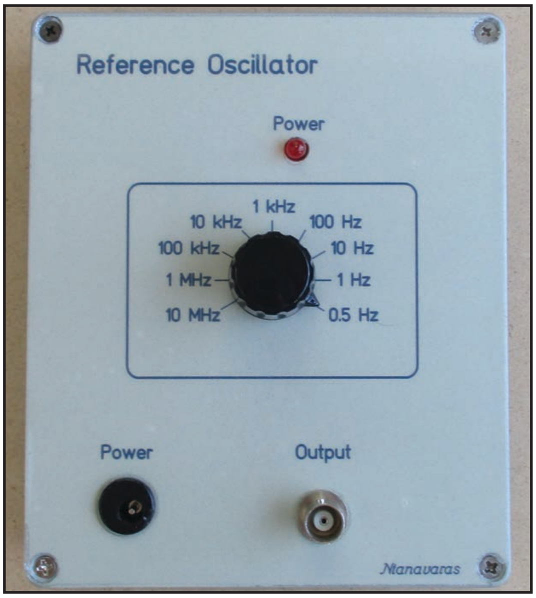

Photo 6: This is the completed Reference Oscillator.

I used an aluminum enclosure to house the PCB and the connectors. Using the "Front Panel Designer" program, I also designed a front panel for the unit. The result was a more professional-looking device that was also easy to operate. I attached the PCB to the front panel using as support the switch, which I placed at the center of the PCB. A BNC connector is used for the output of the oscillator. **Photo 6** shows a view of the completed reference oscillator.

Conclusion

I have found the turntable speed measurement device to be a useful piece of test equipment. It is easy to build it, the cost is reasonable, it measures with excellent accuracy, and it is simple to use. If you don't have this piece of equipment in your lab, here is a good opportunity to build one.

Author's Note: I have a small quantity of PCBs for the construction of the TSM-100 device. Also I can provide a programmed CPU with the proper software. If you are interested, send me an e-mail at gntanavaras@gmail.com

Project Files

To download the TSM-100 Parts List and the Reference Oscillator Parts List, visit http://audioxpress.com/page/audioXpress-Supplementary-Material.html

Sources

Oven-Controlled Crystal Oscillator
Abracon | www.abracon.com

Arduino software
Arduino | www.arduino.cc/en/Main/Software

Eagle PCB design software
Autodesk, Inc. | www.autodesk.com/products/eagle/overview

Front Panel Designer
Schaeffer AG | www.schaeffer-ag.de