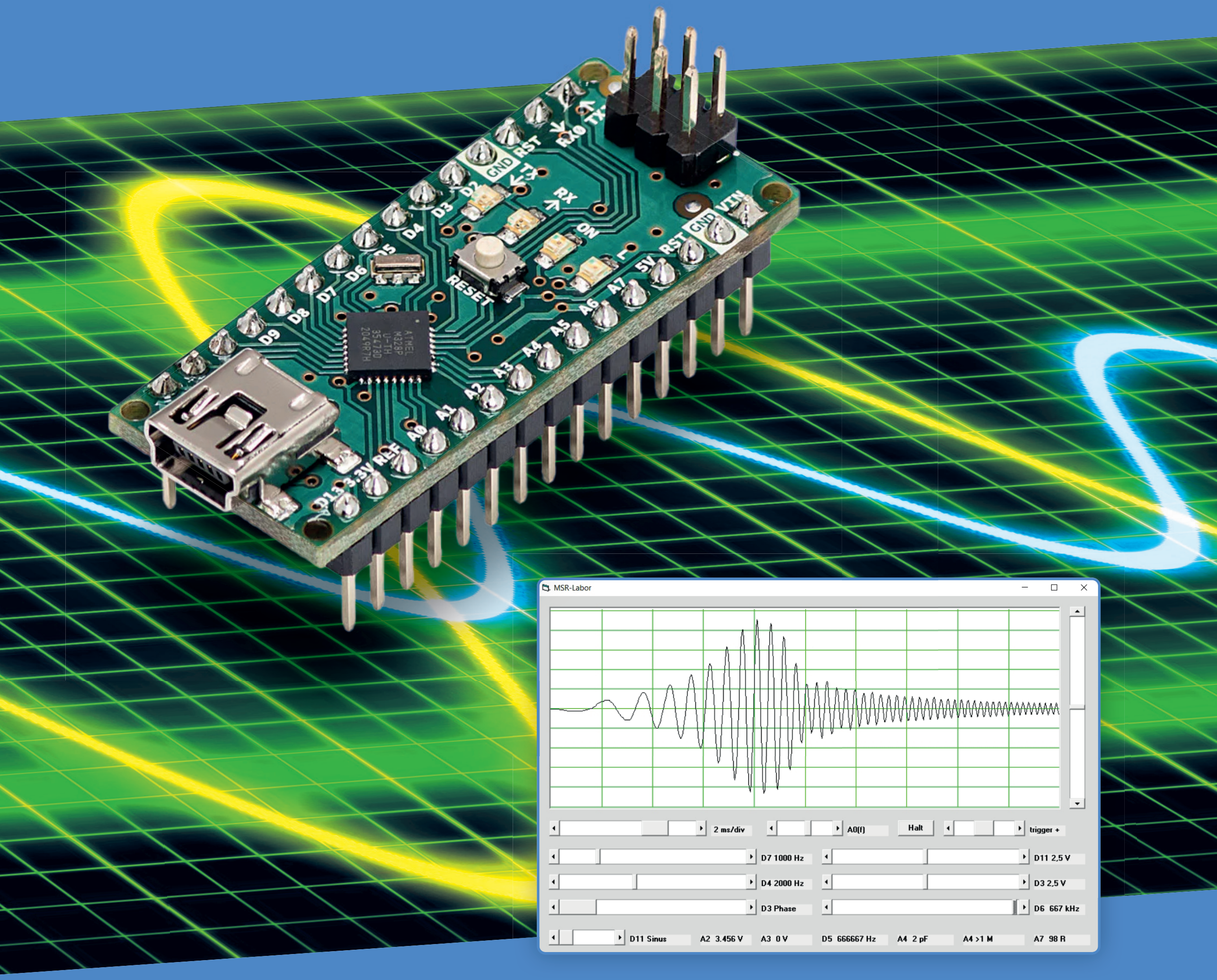
elektor books

The Arduino-Inside Measurement Lab

An 8-in-1 test & measurement instrument for the electronics workbench



2 ms/div
A0(f)
trigger +
D7 1000 Hz
D11 2,5 V
D4 2000 Hz
D3 2,5 V
D3 Phase
D6 667 kHz
D11 Sinus
A2 3.456 V
A3 0 V
D5 666667 Hz
A4 2 pF
A4 >1 M
A7 98 R

Burkhard Kainka

elektor
design > share > earn

The Arduino-Inside Measurement Lab

An 8-in-1 test & measurement instrument for the electronics workbench

Burkhard Kainka

elektor
design > share > earn

● This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
PO Box 11, NL-6114-ZG Susteren, The Netherlands
Phone: +31 46 4389444

● Declaration

The author, editor, and publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, and hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause. All the programs given in the book are Copyright of the Author and Elektor International Media. These programs may only be used for educational purposes. Written permission from the Author or Elektor must be obtained before any of these programs can be used for commercial purposes.

● British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

● **ISBN 978-3-89576-603-9** Print
ISBN 978-3-89576-604-6 eBook

www.elektor.com
Editor: Jan Buiting, MA
Prepress Production: D-Vision, Julian van den Berg
Printers: Ipskamp, Enschede, The Netherlands

Contents

Foreword

A well-equipped electronics lab is crammed with power supplies, measuring devices, test equipment and signal generators. In addition, there are tools, components and the many assemblies and projects you're working on. It can get crowded in the lab. Wouldn't it be better to have one compact device for almost all tasks? All in one, small and versatile, as well as inexpensive and easy to procure.

After several attempts with other systems, the choice fell on the Arduino Nano. On this basis, a PC interface as versatile as possible for measurement and control is to be developed. It simply hangs on a USB cable and, depending on the software, forms the measuring head of a digital voltmeter or PC oscilloscope, a signal generator, an adjustable voltage source, a frequency counter, an ohmmeter a capacitance meter, a characteristic curve recorder and much more.

The circuits and methods collected here are not only relevant for exactly these tasks in the electronics laboratory, but many details can also be used within completely different contexts. Often, you encounter complex tasks in the field of measurement technology during development work with microcontrollers and the methods from this book can be used in these cases, offering a starting point to develop the software in the desired direction.

Stay creative!

Burkhard Kainka

Software and more information about the book:
https://www.b-kainka.de/MeasurementLab.html

Chapter 1 • Preparations

The first considerations for a universal measuring system went in the direction of a digital PC oscilloscope combined with a signal generator. The decisive factor is the achievable cut-off frequency.

1.1 Choice of the Controller

In the first experiment I tested a Bluepill board with a STM32 controller. This Cortex M3 controller with a clock rate up to 72 MHz has a very fast AD converter and an integrated USB interface. Basically, all tasks can be done by the same controller. Analog data is stored while at the same time values of a DDS table are sent to the fast PWM outputs to generate sine signals or other waveforms.

The experiments went quite far, but in the end some interfering factors became known. For the tasks ahead, it is important that the measuring sample and the outputs run in a fast and very accurate time limit. Sampling rates up to 100 kHz at a PWM frequency of 125 kHz with two output and two input channels each were achieved. But there was a permanent slight flutter within the time limit, caused by simultaneous demands of the USB interface. If the actual measurement and signal output is executed within a timer interrupt at 100 kHz, no other interrupt should be allowed. As soon as a second timer interrupt or, as in this case, a USB interrupt is added, the absolutely uniform time grid is over.

In addition, there were delivery problems. The general semiconductor crisis in 2021 meant that the Bluepill board was also hard to come by, so that also ended this way. Bye, bye Bluepill!

There are also other STM controllers. A development board STM32 Nucleo with a Cortex-M0 controller F040R8 with up to 64 MHz was used. The crucial difference was that it has no internal USB interface and transfers its data via the serial interface. The controller transmits its data only when it is measured and is hardly "loaded" by it serial data transmission. In fact, this completely eliminated the signal processing flutter. Actually, the project was on a good path. But I would have had to build a separate board with the STM32 controller and a USB/serial converter, which would have been relatively expensive in the end.

When you enter the 32-bit microcontroller world, the high clock rate is usually the first thing you notice. The rate should allow fast signal processing in real time. However, disappointment often follows as the more complex software seems to defeat the speed advantage. This and the procurement problems made some existing 8-bit controllers more interesting again.

The third attempt employed an 8051-compatible N76E003 from Nuvoton. It has 16 kByte and runs at 16 MHz. Its main advantage is several timers and an on-board PWM unit. This made it possible to achieve a clock rate of 50 kHz, in which both the analog data for the oscilloscope could be measured and the DDS data could be output to the PWM. All time-critical operations would run in a single timer routine.

During further development, the requirements increased because a trigger function had to be included and the gate time of a frequency counter had to be measured at the same time. Therefore, the sampling rate had to be reduced to 32 kHz.

At this point I became curious: What would an ATmega328 do in comparison? First, I tested it with Bascom and was positively surprised right away. Although the controller used in the Arduino has only three timers, it is possible to use one timer at the same time for PWM output as well as for a timer interrupt. The sampling rate is now equal to the PWM output frequency 62.5 kHz. In addition, the serial transmission rate of 1 Mbit/s could be used, making intermediate storage of the measured values unnecessary. This effectively allows the same speed that was achieved with an STM32. In addition, many of the additional functions were proven to fit into the interrupt without consuming too much time.

That was four attempts with the same goal. The winner so far was the ATmega328 on an Arduino Nano. The system is cheap, widely used, and many readers may already have this board. However, this is not necessarily true for Bascom, hence the desire was to work with the Arduino IDE if possible. However, Arduino sketches are not real-time capable, because there are things going on in the background that you don't necessarily see. This is the reason Bascom is much faster than the Arduino-C in most cases.

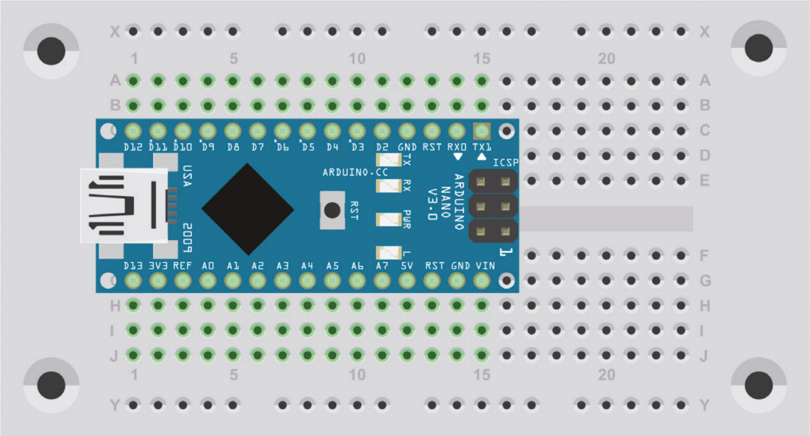
The decisive breakthrough came with the realization that the Arduino IDE can also be used for pure C. This permits using the familiar programming environment with its reliable bootloader and simple operation, while at the same time achieving maximum working speed of the controller.

This establishes the essential cornerstones of the development: An Arduino Nano is supplied with data and energy via its USB cable and returns measured values. Everything is evaluated and displayed on the PC. Programming the PC software with VB6 is only dealt with in passing, whereby the user can fall back on ready-made programs. The book focuses on firmware development in C, optimization of the measurement procedures and typical use of the measurements.

You don't need much more than a solderless breadboard and a Nano. There is still room on the board for an operational amplifier and other components for filters or circuits to be tested. This also demonstrates the correct use of digital measurement devices. With this simple, low sampling rate device, many possible measurement errors can occur in the same way as with a fast DSO. If you work with it on a small scale, you will get fewer surprises on a large scale. In this way, working with the small measurement laboratory doubles as a useful basic course.

1.2 The Arduino Nano

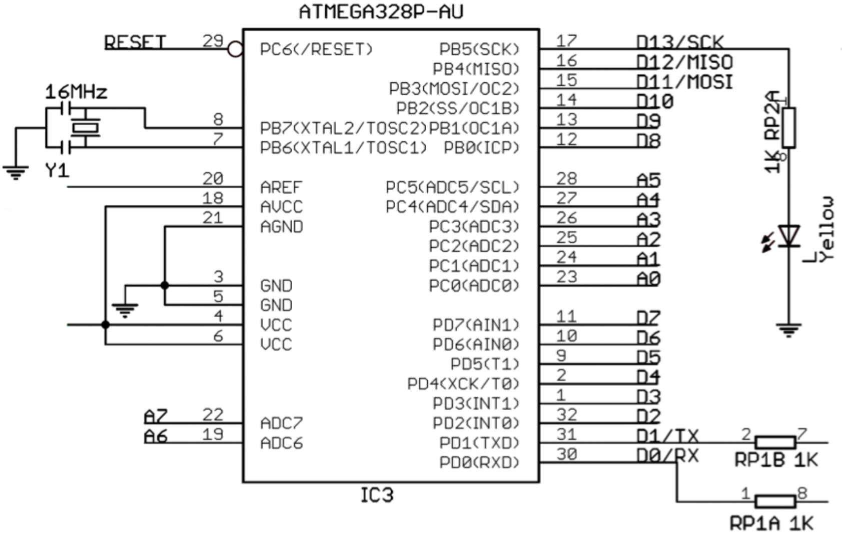
All measurement programs presented in this book run on the Arduino Uno as well as on the Arduino Nano. It only depends on the controller, the ATmega328. The Arduino Nano has the advantage that it can be put on a breadboard together with other components.



Arduino Nano on a breadboard.

Besides the original Arduino Nano, there are several very inexpensive replicas. All of them use the same pinout and can be used for the experiments in this book. They differ in some technical details and often also in the USB socket. Mostly, a USB mini connector is used. It has the advantage of greater mechanical stability, which benefits experimental work. However, USB micro or USB C cables are now more common.

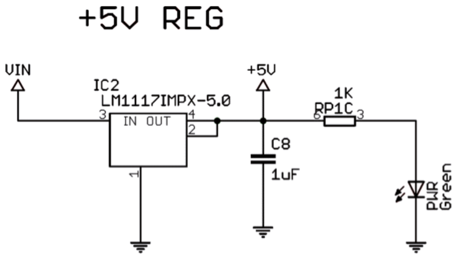
The outer connectors have been renamed to D0 through D13 and A0 through A7 according to Arduino style, and they can be addressed with the Arduino IDE under these names. But for deeper programming it is important to also know the original names of the controller.

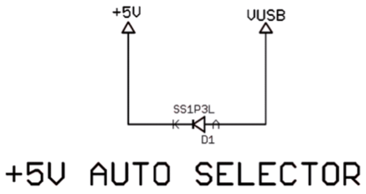


A section from the Arduino Nano circuit diagram.

1.3 Supply Voltage

The original Arduino Nano uses an FT232RL USB converter whose internal voltage regulator also provides the 3.3 V supply voltage to the outside circuitry. Besides there are several replicas which use the cheaper CH340 USB converter. This also has a 3.3 V output but with less current capacity. Some replicas therefore use an additional voltage regulator, while others use the CH340output or leave the 3.3 V output completely free.





Like any Arduino, the Nano can get its 5 V operating voltage either from the USB or from an external voltage source with at least 7 V to Vin. The original Nano uses a Schottky diode in series with the USB supply, whose forward voltage reduces the operating voltage by up to 0.3 V. This diode prevents damage to the PC if a higher voltage is ever accidentally applied to the Nano.

Some Nano replicas omit the series connected Schottky diode in the interest of the full operating voltage. Often, two diodes are connected in antiparallel between VCC and GND. This provides protection against incorrectly polarized voltage, but not against overvoltage at VCC.

"Overvoltage? That can never happen to me", everyone would say. But while authoring this book it happened nevertheless. Because the Arduino Nano was simultaneously programmed as an oscilloscope and as a signal generator, adjustable within wide limits, it should examine the switching behaviour of a reed relay, which was connected to the laboratory power supply outputting 30 V via a switching transistor. There must have been a mistake and the 30 V came directly to the A0 analog input and from there, via the internal protection diode, at the port connector of the ATmega to the VCC line. The Nano survived this almost unscathed, only pin A0 had an internal termination against GND afterwards. But a USB hub and a webcam connected to it at the same time were destroyed. Later, it was found out that the USB hub had a series diode, which probably prevented even bigger damage to the PC.

By the way, the replica version of the Nano mostly used in Franzis Verlag packages would have prevented this damage just like the original Nano, because both have the Schottky diode in series with the USB supply.

Another pitfall lurks with all Nanos: the VIN connection normally remains free when the controller is powered via USB. If you connect it to GND, the 5 V voltage regulator goes into a forbidden state, which can lead to a latch up. The regulator then triggers like a thyristor and can get extremely hot. Also, a relatively large capacitor between VIN and GND can trigger this state at power-up due to its charging current. So this pin should be left free if you don’t want to connect an external power supply.

General precautions mainly concern the power supply used. An external power supply to Vin should only be used in exceptional cases because double caution is then necessary. The safest way is to supply power via the USB port, with an USB hub in between for even more safety. The disadvantage is often a reduced voltage at VCC, where instead of 5.0 V often only 4.5 V are measured. If the USB hub has an optional external power supply, this can be connected if high accuracy is required. VCC also serves as a voltage reference for the AD converter.

If a higher voltage is needed internally, voltage doubling can be used, which is presented in section 8.6. Possible errors are then less perilous because the circuit cannot supply large currents.

All connections to the outside world, for example, to external measuring objects, should be checked particularly carefully. For safety, additional 1 kΩ to 10 kΩ series resistors may be connected in series to test leads, which provide effective current limitation in the event of a fault.

Chapter 2 • Preliminary Tests

At first, quite regular Arduino sketches will to be discussed and used here. They allow you to test the different inputs and outputs, get to know typical measurement methods, and at the same time examine how fast the controller is able to work with them.

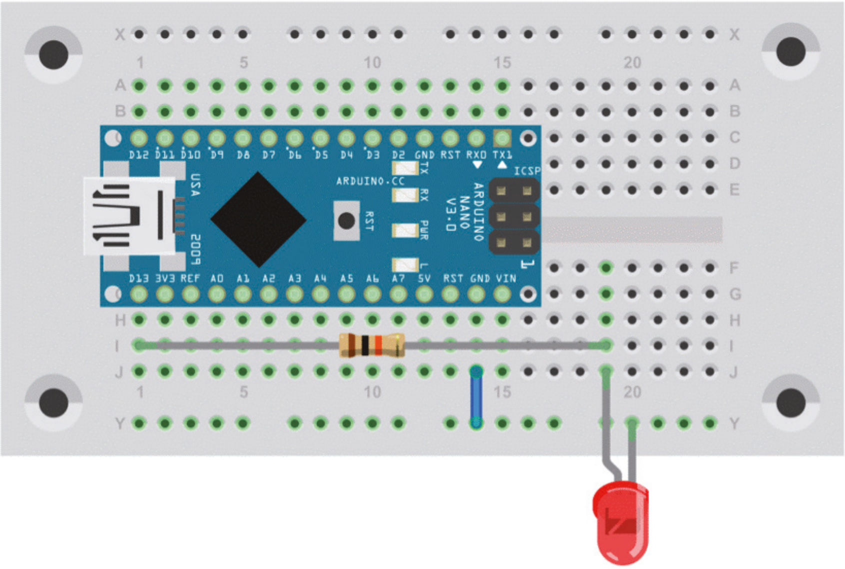
2.1 Port Outputs

Whenever I want to test an Arduino, I first load the program "Blink" from the examples (group 01. Basics).

```
void setup() { pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
 digitalWrite(LED_BUILTIN, HIGH);
 delay(1000);
 digitalWrite(LED_BUILTIN, LOW);
 delay(1000);
}
```

The built-in LED will then flash at a rate of 0.5 Hz, as it is turned on for one second and turned off, also for one second.

The internal LED is connected to pin D13 through a series resistor. Here you can connect an additional external LED with its own series resistor. With modern, high-efficiency LEDs, a low current far under 1 mA is sufficient. Therefore, a 10 kΩ resistor is suggested here.



Additional connection of an external LED.