

INSTRUCTION MANUAL

Advanced Physics Lab

Electronics D

Modern Aspect of Data Taking and Processing with a Microcontroller

Pirmin Berger & Michael Reichmann

Abstract

The experiment “Digital Electronic” provides an introduction into modern data taking by operating simple digital circuits utilising an Arduino board. This manual will inform you about the Arduino board, the installation of the required software and the electrical components you will have to use. Basic knowledge on electronics, how to use oscilloscopes, bread boards and power supplies is recommended.

During the experiment you will learn how to build a circuit that measures the temperature, how to operate it using the Arduino board and to modify and improve it using more components.

In case you should already have previous knowledge we will provide many more material and own ideas on implementation are very welcome and can be built consulting the assistants.

Contents

1	Introduction	2
1.1	Arduino Board	2
1.2	Transistor	2
2	Basics	4
2.1	Arduino Uno	4
2.2	Grove Base Shield	5
2.3	The Software	5
2.3.1	Arduino integrated development environment (IDE)	5
2.3.2	Board Drivers	6
2.4	Programming	6
2.5	Project Management With Git	6
2.6	Thermistor	6
2.7	Common Collector	8
2.8	Operational Amplifier	8
2.9	Voltage Divider	8
3	Setup and Experimental Procedure	9
3.1	Setting up the Arduino	9
3.2	Blinking LED on Bread Board	9
3.3	Grove Temperature Sensor	9
3.4	Building Your Own Temperature Sensor	9
3.5	Building a Heating System	9
3.6	Building a Cooling System	9
3.7	Read Out the Fan Speed (Advanced)	9
3.8	Adding a Display (Advanced)	9
4	Analysis / Protocol	9

1 Introduction

Arduino is a computer company, project and user community based on easy-to-use hardware and software, that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. All products are distributed as open-source hardware and software, and its licences permit the manufacture of Arduino boards and software distribution by anyone. The boards are commercially available in preassembled form, or as do-it-yourself (DIY) kits.

The Arduino project started in 2003 as a program for students without a background in electronics and programming at the Interaction Design Institute in Ivrea (Italy). The aim was to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. The actual name Arduino comes from a bar in Ivrea, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014 [5].

In order to work with the Arduino Boards the Arduino programming language, based on Wiring, and the Arduino Software (IDE), based on the Processing are used [1]. Both Wiring and Processing are programming languages using a simplified dialect of features from the programming languages C and C++.

1.1 Arduino Board

The original boards were produced by the Italian company Smart Projects but as of 2018, 22 versions of the Arduino hardware have been commercially produced. The information and specifications of these boards can be found on this [website](#). During this Lab you will work the Arduino Uno shown in Figure 1.

The Arduino Boards use a variety of microprocessors and controllers and are equipped with sets of digital and analogue input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits. The boards feature serial communications interfaces, Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.

Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell the board what to do by sending a set of instructions to the microcontroller.

1.2 Transistor

The invention of the transistor was announced in 1948 by the American physicists, J. Bardeen and W. H. Brattain as a new type of amplifying device made from semiconducting crystals. At that time almost no one could have foreseen the revolutionary developments that were to follow, developments so important and far-reaching as to change the whole outlook of the

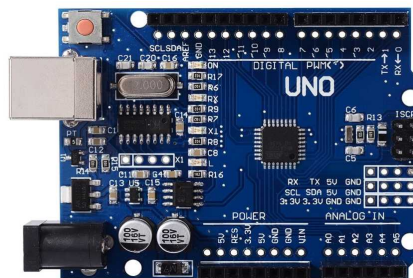


Figure 1: Arduino Uno.

science and technology of electronics. The physical principles of a transistor had been worked out in conjunction with their colleague, W. Shockley. In recognition of their work the three physicists were awarded jointly the Nobel Prize for Physics in 1956.



(a) Point-contact transistor.



(b) Bardeen, Brattain and Shockley.

The term “transistor” is a combination from the words *transformer* and *resistor*, since the device is made from resistor material and transformer action is involved in the operation. In the beginning only point-contact transistors existed, but due to their vulnerability to mechanical shock they were soon replaced by junction transistors which are firmly established now [3].

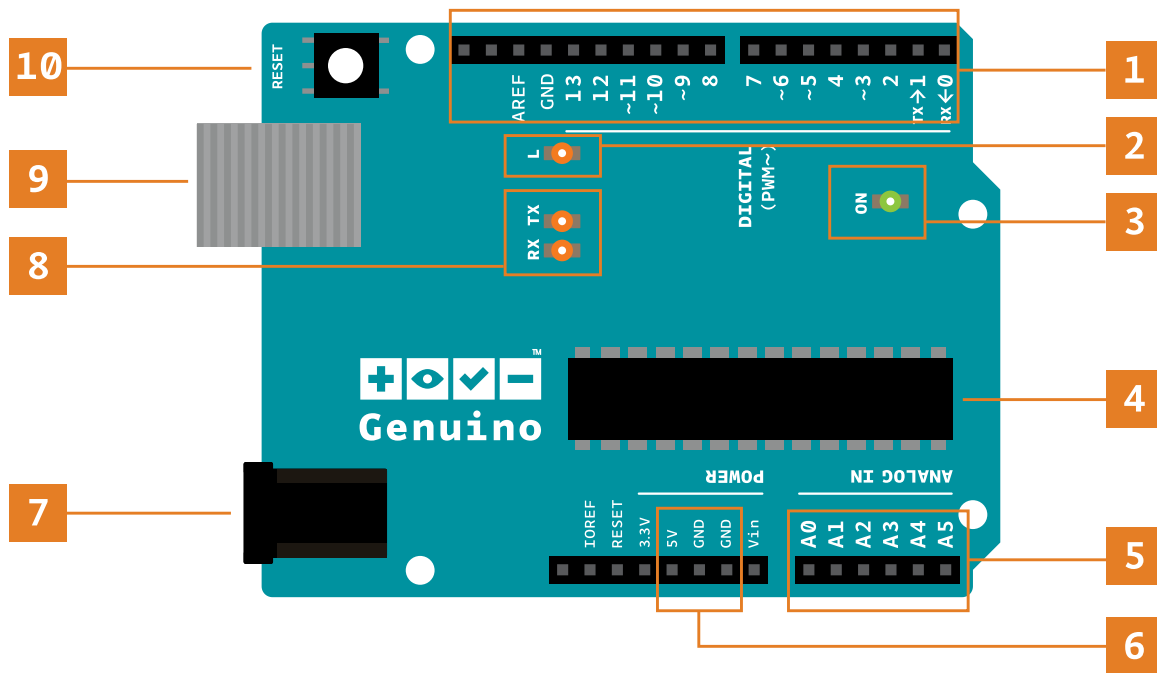
The transistor is the key active component in practically all modern electronics. It is considered as one of the greatest inventions of the 20th century. Its importance in today’s society rests on its ability to be mass-produced using a highly automated process that achieves astonishingly low per-transistor costs (10 femto\$/transistor) [2].

Although billions of individually packaged (discrete) transistors are produced every year the vast majority of transistors are now produced in integrated circuits (ICs). A logic gate consists of up to about twenty transistors whereas an advanced microprocessor, as of 2009, can use as many as 3 billion transistors. In 2014, about 10 billion transistors were built for each single person on Earth [2].

2 Basics

This section will give you the basic information about the components we are using in this lab.

2.1 Arduino Uno



1. **Digital pins:** used with `digitalRead()`, `digitalWrite()`, and `analogWrite()` methods, `analogWrite()` only works on pins with the PWM symbol
2. **Pin 13 LED:** only built-in actuator
3. **Power LED**
4. **ATmega microcontroller**
5. **Analogue in:** used with `analogWrite()` method
6. **GND and 5V pins:** provide 5 V power and ground (GND) to the circuits
7. **Power connector:** additional power supply, accepted voltages: 7 V to 12 V
8. **TX and RX LEDs:** indicate communication between Arduino and computer
9. **USB port:** used for powering and communication with computer
10. **Reset button:** resets the ATmega microcontroller

2.2 Grove Base Shield

The so called shields are printed circuit expansion boards, which plug into the normally supplied Arduino pin headers. The Grove Base Shield is one example that simplifies projects that require a lot of sensors or LEDs. With the Grove connectors on the base board, one can add all the Grove modules to the Arduino Uno very conveniently.

[41][.51].25grovegrove1side viewtop view[Grove shield.]

There are 16 Grove connectors on the Base Shield which are shown in Table 1. Apart from the connectors the board also consists of a reset (RST) button, a green LED to indicating power status, a toggle switch and four rows of pinouts, which is equivalent to the pinout of the Arduino.

Specification	Name	Quantity
Analog	A0/A1/A2/A3	4
Digital	D2/D3/D4/D5/D6/D7/D8	7
UART	UART	1
I2C	I2C	4

Table 1: Base Shield connectors.

Every Grove connector has four wires, one of which is the voltage common collector (VCC). Since some micro-controller main boards need need different supply voltages the power toggle switch allows you to select the suitable voltage. In the case of the Arduino Uno a voltage of 5 V is required [4].

2.3 The Software

2.3.1 Arduino IDE

All you require to write programs and upload them to your board is the Arduino Software (IDE). There are two options how to use it:

1. Online IDE

- no installation required
 - needs plugin if you want to upload sketches from Linux
- requires you to create account with e-mail verification
- save sketches in cloud (available from all devices)
- always most up-to-date version
- instructions on the website

2. Desktop IDE

- if you want to work offline

- installation usually very straightforward and in has general no dependencies
- if you need help, follow installation instructions depending on your operating system (OS)
 - [Linux](#)
 - [Mac OS X](#)
 - [Windows](#)

2.3.2 Board Drivers

First the Arduino board has to be connected to the computer via the USB cable which will power the board indicated by the green power (PWR) LED. The board drivers should then install automatically in Linux, Mac OS X and Windows. If the board was not properly recognised, follow these [instructions](#).

2.4 Programming

In order to get a feeling for the programming language it is recommended to have a look at the examples first which can be found under: **File > Examples**

A list of the most common methods is shown in Table 2. For more information look at the [detailed description](#).

2.5 Project Management With Git



Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files [6].

2.6 Thermistor

If, for a given temperature, the current is directly proportional to the applied voltage the electrical component is said to obey Ohm's law. Such components are called linear resistors. If a component does not meet this requirement it is termed a non-linear resistor, which falls into two classes – the temperature-sensitive type and the voltage-sensitive type. The temperature-sensitive types are often known as thermistors and change the resistance very reproducibly. The word is a portmanteau of *THERM*ally-sensitive and res**ISTOR**.

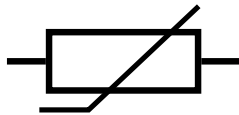
Category	Method Syntax	Description
Digital I/O	<code>digitalRead(pin)</code>	reads the value from the digital pin (HIGH or LOW)
	<code>digitalWrite(pin, value)</code>	writes HIGH or LOW value to the digital pin
	<code>pinMode(pin, mode)</code>	configures the pin as INPUT or OUTPUT
Analogue I/O	<code>analogRead(pin)</code>	reads the value (0-1023) from the pin
	<code>analogWrite(pin, value)</code>	writes an analogue value (PWM wave) to the pin
	<code>analogReference(type)</code>	configures the reference voltage used for analogue input
Advanced I/O	<code>tone(pin, f, duration)</code>	generates a square wave of frequency f [Hz] for a duration [ms]
	<code>pulseIn(pin, value)</code>	returns the time [ms] of a pulse, if value is HIGH: waits until HIGH and stops when LOW
Time	<code>delay(time)</code>	pauses the program for a time [ms]
	<code>micros()</code>	returns time since starting the program [μ s]
	<code>millis()</code>	returns time since starting the program [ms]
Math	<code>constrain(x, a, b)</code>	constrains a number x to be in range [a, b]
	<code>map(x, a, b, c, d)</code>	re-maps x from range [a, b] to range [c, d]
	<code>random(a, b)</code>	returns pseudo-random number in range [a, b]
	<code>abs(value)</code>	return the absolute value
	<code>:</code>	further general math commands
Serial	<code>Serial.begin(speed)</code>	initialise serial communication at speed [bit/s]
	<code>Serial.print(value)</code>	prints the value to the serial port
	<code>Serial.println(value)</code>	prints the value to the serial port with <code>[\r\n]</code>
	<code>Serial.read()</code>	reads incoming serial data

Table 2: Most common methods for controlling the Arduino board and performing computations.

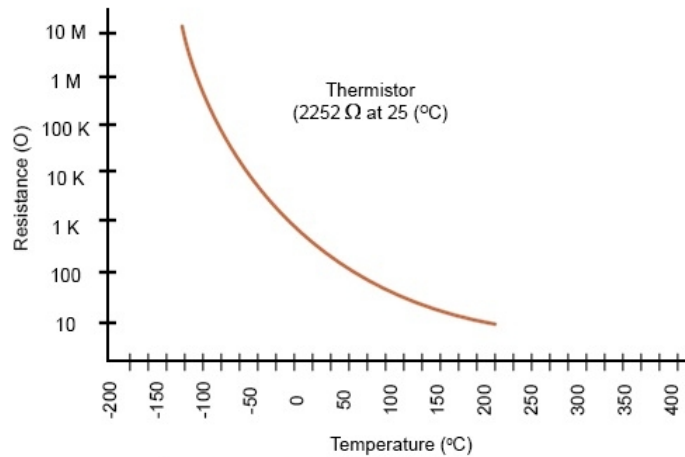
They consist of the sintered oxides of manganese and nickel with small amounts of copper, cobalt or iron added to vary the properties and the physical shape is usually a bead, rod or a disc. The electronic symbol is shown in Figure 3a. The resistance is given by

$$R = R_0 \cdot e^{-B\left(\frac{1}{T_0} - \frac{1}{T}\right)} \quad (1)$$

where R_0 and B are constants depending upon the composition and physical size, T is the temperature in $^{\circ}\text{K}$, and T_0 the absolute zero. Thermistors can be classified into two types, depending on the classification of B . If B is positive, the resistance increases with increasing temperature, and the device is called a positive temperature coefficient (PTC) thermistor, or posistor. If B is negative, the resistance decreases with increasing temperature, and the device is called a negative temperature coefficient (NTC) thermistor.



(a) Electronic symbol.



(b) NTC resistance/temperature characteristic.

Figure 3: Thermistor properties

Thermistors have very widespread applications as thermometers

During this lab you will work with the NTC 100 k Ω [B57164-K104-J](#) thermistor.

2.7 Common Collector

2.8 Operational Amplifier

Should be covered in AP next semester

2.9 Voltage Divider

Should be covered in AP next semester

3 Setup and Experimental Procedure

3.1 Setting up the Arduino

3.2 Blinking LED on Bread Board

3.3 Grove Temperature Sensor

3.4 Building Your Own Temperature Sensor

3.5 Building a Heating System

3.6 Building a Cooling System

3.7 Read Out the Fan Speed (Advanced)

3.8 Adding a Display (Advanced)

4 Analysis / Protocol

List of Acronyms

IDE integrated development environment

DIY do-it-yourself

USB Universal Serial Bus

I/O input/output

GND ground

RST reset

PWR power

IC integrated circuit

VCC voltage common collector

OS operating system

PTC positive temperature coefficient

NTC negative temperature coefficient

References

- [1] Arduino. Introduction to Arduino. <https://www.arduino.cc/en/Guide/Introduction>, 2018. [Online; accessed 15-May-2018].
- [2] Dan Hutcheson. A look at Moore’s Law. <https://spectrum.ieee.org/computing/hardware/transistor-production-has-reached-astronomical-scales>, 2018. [Online; accessed 15-May-2018].
- [3] G.H. Olsen. *Electronics: A General Introduction for the Non-Specialist*. Springer US, 2013.
- [4] Seeed. Base Shield V2. <https://www.seeedstudio.com/Base-Shield-V2-p-1378.html>, 2018. [Online; accessed 15-May-2018].
- [5] Wikipedia contributors. Arduino — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Arduino&oldid=839287116>, 2018. [Online; accessed 15-May-2018].
- [6] Wikipedia contributors. Git — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Git&oldid=840401990>, 2018. [Online; accessed 16-May-2018].