

Jak programovat Tinykeyer?

Na úvod: veškeré operace dělám v Linuxu, jak je to ve Windows, se jen domnívám. Zásady jsou ale všude stejné...

PŘÍPRAVA



Nejdřív musíte vědět, na kterém portu Arduino vlastně máte. V Linuxu je to jednoduché, můžete použít skript `devcheck.sh`. U Windows je to složitější – systém sice pozná, že bylo přidáno nějaké USB zařízení, ale bude třeba instalovat ovládač, aby se vytvořil virtuální COM port. Skript i ovládače najdete na <http://www.ok1rr.com/tinykeyer/index.html>.

Než začneme, je nutné opatřit si vývojové prostředí (IDE) pro Arduino. Najdete ho ke stažení zdarma na:

<https://www.arduino.cc/en/Main/Software>



POZOR! Nepoužívejte běžnou (regulérní) verzi, stáhněte si tzv. HOURLY BUILD!



Regulérní verze Arduino IDE vykazují problémy s kompilací kódu, což **není zaviněno kódem**, ale právě tím IDE, které z neznámých důvodů neobsahuje všechny komponenty, nutné k ověření, kompilaci a zavedení kódu do Arduina. Jsou zde problémy se závislostmi a kompilace neproběhne až do konce, IDE hlásí chyby atd.

HOURLY BUILD najdete na zmíněné stránce poněkud níž. Je viditelně označený a je k dispozici pro všechny běžné operační systémy, tj. Windows, Mac OS X (Mac OSX Lion a pozdější verze), Linux 32 bit, Linux 64 bit a Linux ARM (experimentální). Jde o poměrně velký balík, stahování je přibližně 149 MB (Windows) nebo 94 MB (Linux)! Pro Windows a Mac OS X je k dispozici archiv .zip, pro linuxové platformy archiv tar.xz.

The screenshot shows the Arduino website interface. At the top, there is a navigation bar with links like 'Buy', 'Download', 'Products', 'Learning', 'Forum', 'Support', and 'Blog'. The main content area features a large section for 'ARDUINO 1.6.12' with a description and a 'Get' button for the Windows app. Below this, there is a section for 'HOURLY BUILDS' which is highlighted with a red box. A red arrow points from the text 'TO JE ONO!' to this section. The 'HOURLY BUILDS' section includes a 'LAST UPDATE' timestamp and a list of operating systems: Windows, Mac OS X (Mac OSX Lion or later), and Linux (32 bit, 64 bit, ARM experimental). A red box also highlights the 'Linux 32 bit, Linux 64 bit, Linux ARM (experimental)' text, with a red arrow pointing to it from the text 'vybrat podle vašeho systému'.

Další postup byl prováděn na platformě Linux 64 bit, ale bude stejný pro všechny systémy (pro Windows ověřeno).

Archiv rozbalíme nejlépe přímo do hlavního adresáře, kde se automaticky vytvoří podadresář **arduino-nightly**.

Windows

Název	Velikost	Komprimovaný	Režim	CRC	Metoda	Datum
arduino-nightly	9 složek, 9 souborů			Store	21.10.16 18:34
drivers	5 složek, 12 souborů			Store	21.10.16 18:33
examples	11 složek			Store	21.10.16 18:34
hardware	2 složky, 3 soubory			Store	21.10.16 18:34
java	2 složky, 7 souborů			Store	21.10.16 18:34
lib	2 složky, 59 souborů			Store	21.10.16 18:34
libraries	18 složek			Store	21.10.16 18:34
reference	4 složky			Store	21.10.16 18:34
tools	2 složky, 1 soubor			Store	21.10.16 18:34
tools-builder	1 složka			Store	21.10.16 18:34
arduino-builder.exe	3,7 MiB	1,2 MiB	DEC4ABDC	Deflate	21.10.16 18:34
arduino.exe	393,0 KiB	70,1 KiB	8916F7F6	Deflate	21.10.16 18:34
arduino.l4j.ini	20 B	20 B	AA0F5D25	Deflate	21.10.16 18:34
arduino_debug.exe	390,5 KiB	68,8 KiB	AB2D0E3D	Deflate	21.10.16 18:34
arduino_debug.l4j.ini	20 B	20 B	AA0F5D25	Deflate	21.10.16 18:34
libusb0.dll	42,5 KiB	18,7 KiB	214C4521	Deflate	21.10.16 18:33
msvcpr100.dll	411,3 KiB	131,5 KiB	38F12A25	Deflate	21.10.16 18:34
msvcpr100.dll	752,3 KiB	401,8 KiB	DB505E33	Deflate	21.10.16 18:34
revisions.txt	78,6 KiB	27,0 KiB	7612ABC6	Deflate	21.10.16 18:33

Linux

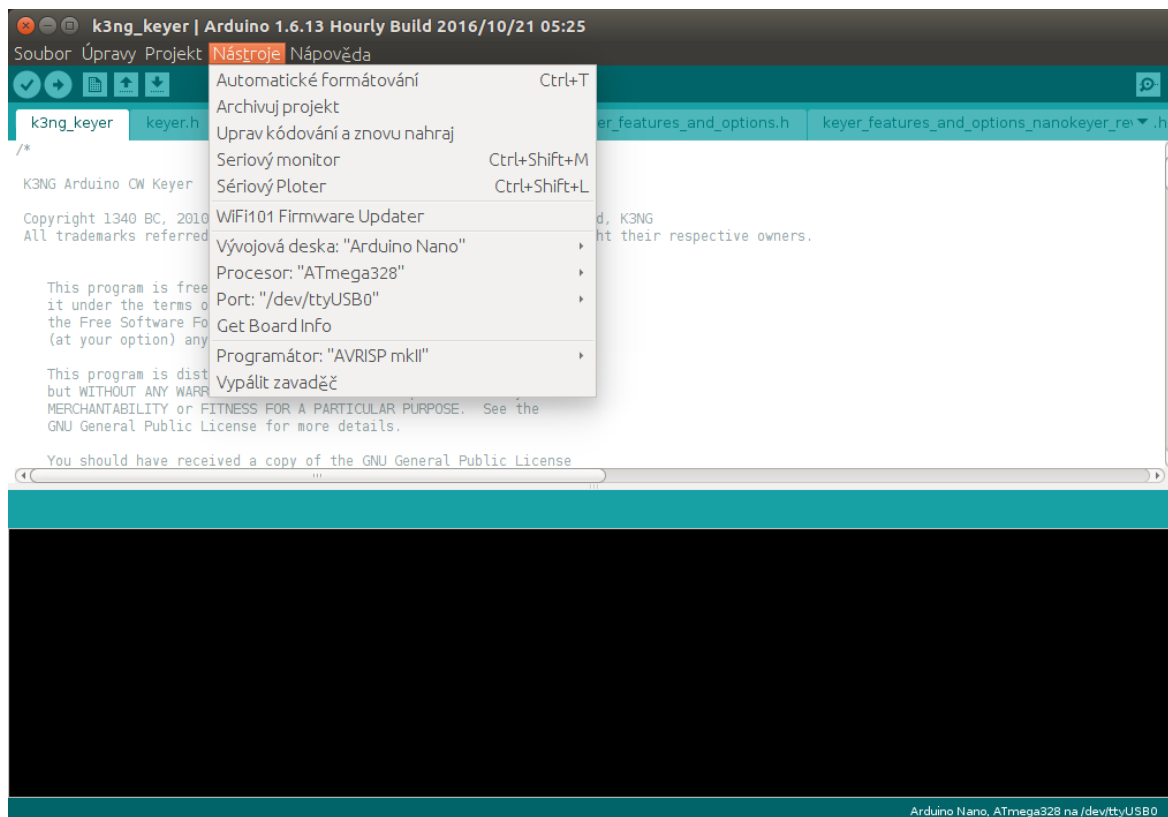
Název	Velikost	Vlastník	Skupina	Datum
arduino-nightly	8 složek, 5 souborů	jenkins	jenkins	21.10.16 17:26
examples	11 složek	jenkins	jenkins	21.10.16 17:25
hardware	2 složky, 3 soubory	jenkins	jenkins	21.10.16 17:26
java	4 složky, 7 souborů	jenkins	jenkins	01.04.16 6:17
lib	2 složky, 62 souborů	jenkins	jenkins	21.10.16 17:26
libraries	18 složek	jenkins	jenkins	21.10.16 17:25
reference	4 složky	jenkins	jenkins	21.10.16 17:25
tools	2 složky, 1 soubor	jenkins	jenkins	21.10.16 17:25
tools-builder	1 složka	jenkins	jenkins	26.08.16 18:59
arduino	946 B	jenkins	jenkins	21.10.16 17:25
arduino-builder	3,9 MiB	jenkins	jenkins	21.10.16 17:26
install.sh	9,0 KiB	jenkins	jenkins	21.10.16 17:25
revisions.txt	76,9 KiB	jenkins	jenkins	21.10.16 17:25
uninstall.sh	86 B	jenkins	jenkins	21.10.16 17:25

Na obrázku je zároveň vidět, jaké adresáře a soubory obsahuje nově vzniklý podadresář **arduino-nightly**. Archiv je kompletní do té míry, že vlastní instalace je jen pouhé zavedení položek a ikon do menu, příp. na pracovní plochu. V Linuxu spustíme skript *install.sh* a vše potřebné se vytvoří, ve Windows stačí jen udělat zástupce **arduino.exe** a umístit na plochu, můžeme samozřejmě přímo spouštět **arduino.exe** ze správce souborů. Po rozbalení archivu příp. instalaci spustíme IDE (ověříme si, že jde spustit). IDE Arduina je automaticky v češtině, pozná *locale* systému (příp. českou verzi Windows). Přitom dojde i vytvoření podadresáře **sketchbook**, kde budou umístěny soubory s naším firmware. Jak to vypadá u mě, ukazuje následující obrázek, podstatné je na tom to, že v podadresáři **sketchbook** musíte mít další podadresář **k3ng_keyer**. Pokud tam je podadresář **libraries** příp. další soubory, patří to k samotnému vývojovému prostředí a nechte je tam, kde jsou. **k3ng_keyer** má svoje **libraries**, které tam už jsou.

Název	Přípona	Velikost	Změněno	rw
..			<DIR>	
libraries			<DIR> 2.10.2016 17:24	rwX
k3ng_keyer			<DIR> 1.11.2016 18:00	rwX

Adresář **k3ng_keyer** je vlastně obsahem firmware, které si stáhnete. Firmware má vždy název **tinykeyer_fwxxxxxx.zip**, kde číslo (xxxxxx) udává verzi, vlastně je to datum jeho uveřejnění (např. **tinykeyer_fw161101.zip**). Archiv firmware **tinykeyer_fwxxxxxx.zip** tedy rozbalíme do adresáře **sketchbook**. O nic jiného se nemusíte starat.

Dále v IDE v menu **Nástroje** nastavíme typ desky. Pro Tinykeyer to **vždy musí být Arduino Nano** a procesor **Atmega 328**.



Port vždy nastavíme ten, o kterém víme, že je na něm naše Arduino. V Linuxu je to vždy /dev/ttyUSBx, ve Windows bude mít port označení COMx (x je číslo). Tím je příprava k programování hotová.

PROGRAMOVÁNÍ

Kvůli programování je třeba klíč vyjmout z krabičky, je totiž **nutné vytáhnout jumper JP1**. Vyšroubujte čtyři šroubky v rozích čelního panelu a povolte matky na konektorech na zadní straně. Desku klíče vytáhněte dopředu (však víte, vždyť jste ho stavěli!).



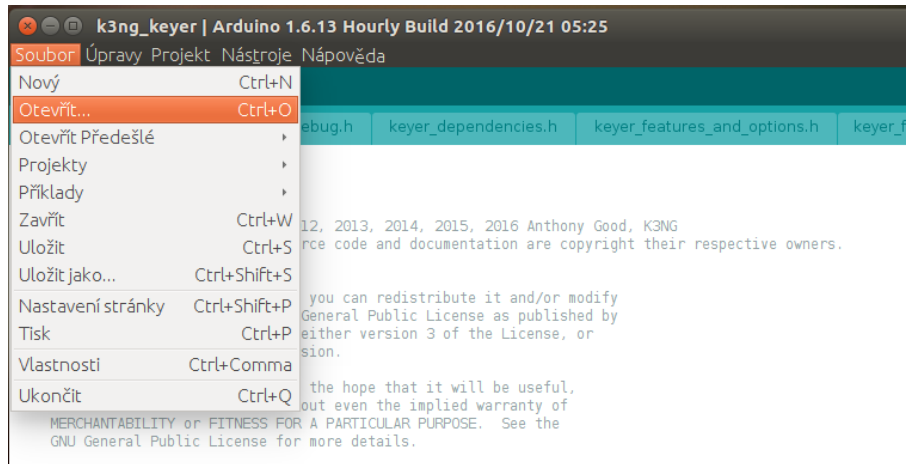


Při programování musí být jumper vytažený, při provozu musí být na místě!

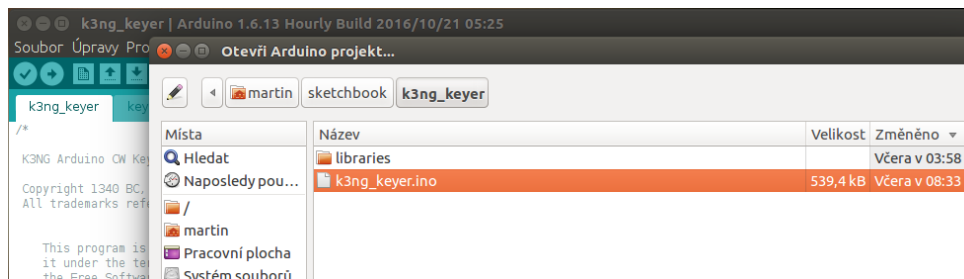


Klíč připojte k USB portu. Pokud byl klíč již naprogramován předtím, „pozdraví“ HI.

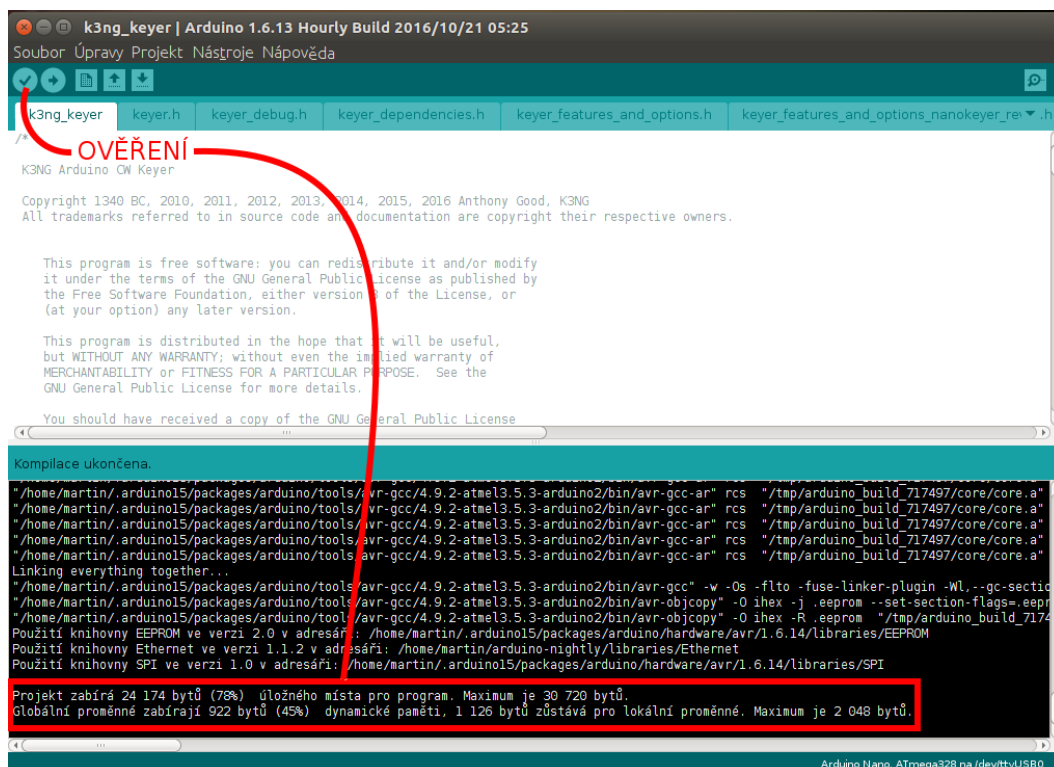
V IDE Arduina je třeba otevřít soubor firmware.



Objeví se dialog, ve kterém „navigujeme“ na soubor **k3ng_keyer.ino**.



Po otevření **k3ng_keyer.ino** se rovněž otevřou i všechny související soubory. Ověříme správnost a bezchybnost kódu prvním tlačítkem s fajfkou.



Pokud je vše v pořádku, objeví se výsledek s oznámením o velikosti spustitelného souboru, procentech zabrané paměti a další údaje. Můžete tedy začít s nahráváním souboru do Arduina. Dělá se to v pořadí druhým tlačítkem se šipkou.

```
k3ng_keyer | Arduino 1.6.13 Hourly Build 2016/10/21 05:25
Soubor Úpravy Projekt Nástroje Nápověda

k3ng_keyer keyer.h keyer_debug.h keyer_dependencies.h keyer_features_and_options.h keyer_features_and_options_nanokeyer_rei...h

/*
K3NG Arduino OW Keyer

Copyright 1340 BC, 2010, 2011, 2012, 2013, 2014, 2015, 2016 Anthony Good, K3NG
All trademarks referred to in source code and documentation are copyright their respective owners.

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or

Konec nahrávání.

Programmer Type : Arduino
Description      : Arduino
Hardware Version: 2
Firmware Version: 1.16
Vtarget         : 0.0 V
Varef           : 0.0 V
Oscillator      : Off
SCK period      : 0.1 us

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "/tmp/arduino_build_175813/k3ng_keyer.ino.hex"
avrdude: writing flash (24174 bytes):

Writing | ##### | 100% 7.37s

avrdude: 24174 bytes of flash written
avrdude: verifying flash memory against /tmp/arduino_build_175813/k3ng_keyer.ino.hex:
avrdude: load data flash data from input file /tmp/arduino_build_175813/k3ng_keyer.ino.hex:
avrdude: input file /tmp/arduino_build_175813/k3ng_keyer.ino.hex contains 24174 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 5.70s

avrdude: verifying ...
avrdude: 24174 bytes of flash verified

avrdude done. Thank you.

16 Arduino Nano, ATmega328 na /dev/tty/USB0
```

Pokud vše proběhne v pořádku, objeví se výsledek jako na obrázku a klíč po dokončení ukládání opět „pozdraví“ HI.

MOŽNÉ CHYBY

```
k3ng_keyer | Arduino 1.6.13 Hourly Build 2016/10/21 05:25
Soubor Úpravy Projekt Nástroje Nápověda

k3ng_keyer keyer.h keyer_debug.h keyer_dependencies.h keyer_features_and_options.h keyer_features_and_options_nanokeyer_rei...h

/*
K3NG Arduino OW Keyer

Copyright 1340 BC, 2010, 2011, 2012, 2013, 2014, 2015, 2016 Anthony Good, K3NG
All trademarks referred to in source code and documentation are copyright their respective owners.

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License

vybraný seriový port neexistuje a nebo váš board není připojen Okopírovat chybové zprávy

Varef           : 0.0 V
Oscillator      : Off
SCK period      : 0.1 us

avrdude: stk500_set_extended_parms(): protocol error, expect=0x14, resp=0x10
avrdude: stk500_initialize(): failed
avrdude: initialization failed, rc=-1
Double check connections and try again, or use -F to override
this check.

avrdude done. Thank you.

vybraný seriový port
neexistuje a nebo váš board není připojen

Arduino Nano, ATmega328 na /dev/tty/USB0
```

Takhle to vypadá, když nemáte zvolený správný port.

```
k3ng_keyer | Arduino 1.6.13 Hourly Build 2016/10/21 05:25
Soubor Úpravy Projekt Nástroje Nápověda

k3ng_keyer keyer.h keyer_debug.h keyer_dependencies.h keyer_features_and_options.h keyer_features_and_options_nanokeyer_re...h

K3NG Arduino CW Keyer

Copyright 1340 BC, 2010, 2011, 2012, 2013, 2014, 2015, 2016 Anthony Good, K3NG
All trademarks referred to in source code and documentation are copyright their respective owners.

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License

Nahrávám...

/home/martin/.arduino15/packages/arduino/tools/avrdude/6.3.0-arduino6/bin/avrdude -C/home/martin/.arduino15/packages/arduino/tools/avrdude/6.3
avrdude: Version 6.3, compiled on Sep 12 2016 at 15:21:49
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2014 Joerg Wunsch

System wide configuration file is "/home/martin/.arduino15/packages/arduino/tools/avrdude/6.3.0-arduino6/etc/avrdude.conf"
User configuration file is "/home/martin/.avrduderc"
User configuration file does not exist or is not a regular file, skipping

Using Port : /dev/ttyUSB0
Using Programmer : arduino
Overriding Baud Rate : 57600
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x00

Arduino Nano, ATmega328 na /dev/ttyUSB0
```

**Tohle znamená, že nedošlo k ukládání do Arduina.
Nejčastější chybou bývá, že zapomenete vytáhnout jumper JP1.**

Pokud je vše v pořádku, odpojte klíč, vraťte jumper JP1 na své místo a můžete klíč vrátit do krabičky.



**Při programování musí být jumper vytažený,
při provozu musí být na místě!**



Programování je ve skutečnosti velmi jednoduché a snadné, když máte trochu zkušenosti, trvá to asi 20 vteřin. Prakticky není možné nic zkazit, klíč je téměř nemožné „umrtvit“. Tento návod jsem napsal spíš kvůli těm, kteří mají strach se do toho pustit. Obavy opravdu nejsou na místě a pokud se objeví nová verze firmware, neváhejte a aktualizujte, ať se nepřipravíte o nějakou novou vychytávku nebo funkci.