u-nex User Guide

u-nex User Guide

Contents

u-nex User Guide

Introduction

The u-nex (micro-nex) is a compact arduino compatible board designed from the ground up to give you maximum possible functionality in a small form factor and low cost. It is based on the Atmega328p micro-controller and has an on-board USB to serial converter (the FTDI chip – FT232RL).

The u-nex also has an onboard 5V low-drop-out regulator which can deliver a maximum current of 0.8A. It can also be powered using the onboard USB mini connector.

The board ships with the arduino boot loader installed and blinks the LED tagged as LD with an on/off timer of 1 second. The u-nex can be directly used as an arduino compatible board out-of-the-box as we shall soon see. The board can also be programmed using a standard ICSP device to erase the arduino bootloader and use the entire 32KB of flash space.

Specifications

U-nex specifications:
- MCU: Atmega328p
- Frequency: 16MHz
- Input voltage range: 7-12V
- Bootloader: ATmegaBOOT_168_atmega328.hex
- Power supply: mini USB or external supply
- Dimensions: 17.78mmx35.00mm

Technical details:
- Hardware SPI & I2C
- External power or USB power
- Breadboard compatible
- On board 5V,0.8A regulator, with supply to breadboard
- 14 Digital I/O Pins (of which 6 provide PWM output)
- 8 Analog Input Pins
- External ADC reference pin
- Power LED, serial LEDs, user configurable LED
- Compatible with Arduino IDE for Windows/Mac/Linux
- 32KB of programmable flash memory (30KB program code space if using Arduino boot-loader)
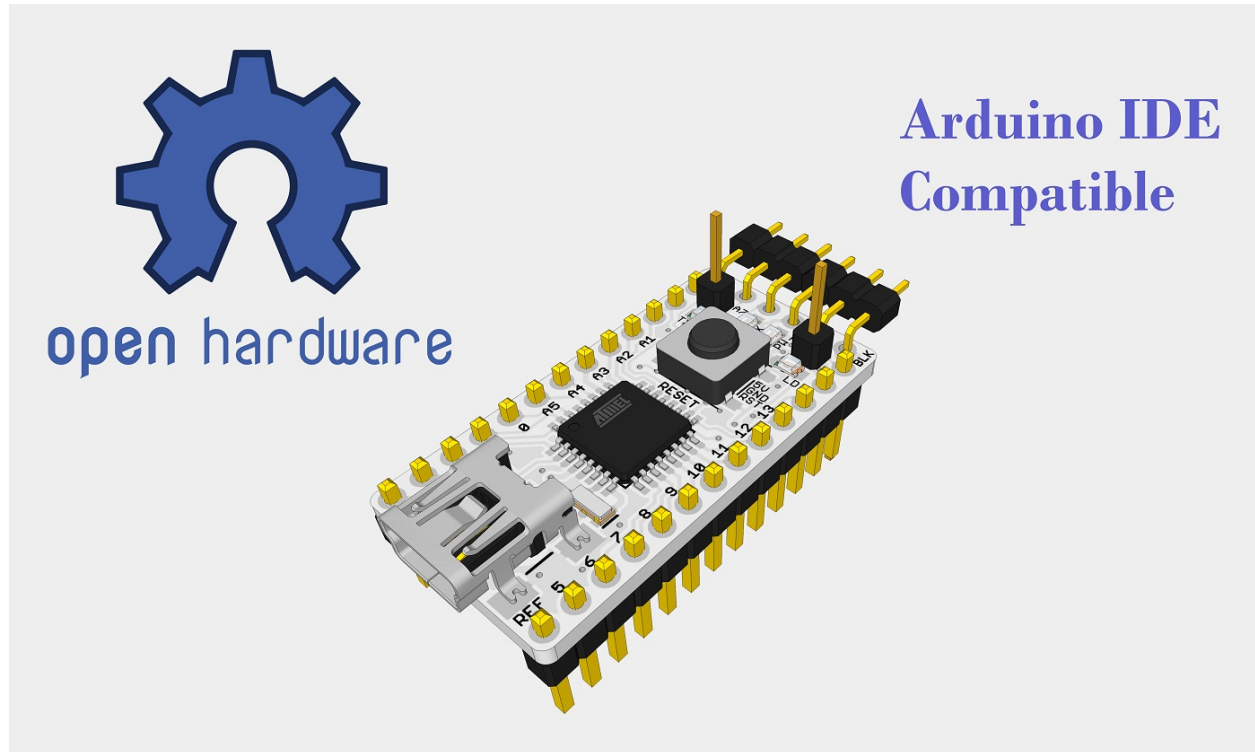- SRAM 2KB
- EEPROM 1KB

Figure 1 Open Hardware & Arduino Compatible
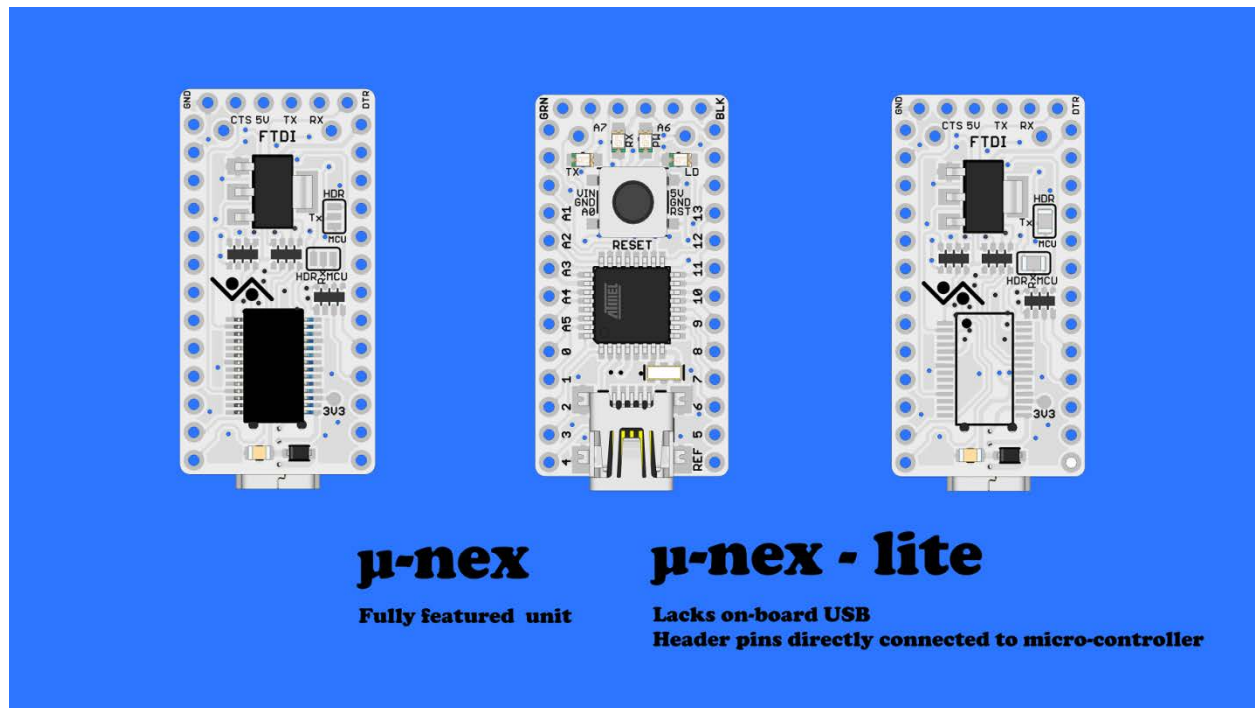
u-nex VS u-nex - lite

Figure 2 u-nex VS u-nex - lite

The u-nex is currently available in two variants, the u-nex and the u-nex-lite. The u-nex features all the functionality promised while the u-nex-lite is a lower cost alternative which only lacks onboard USB. The pin outs and other features are still present.

When shipped the solder jumpers on the u-nex connect the serial port of the Atmega328p to the onboard FTDI chip, while for the u-nex-lite the serial port is connected to the external header pins, enabling you to program it using an external FTDI chip or u-nex.

Pin Mapping:

Atmega328p Pins

Atmega328p	Arduino/u-nex Pin	Functionality
PD0	D0	RXI
PD1	D1	TXO
PD2	D2	
PD3	D3	PWM
PD4	D4	
PD5	D5	PWM
PD6	D6	PWM
PD7	D7	
PB0	D8	
PB1	D9	PWM
PB2	D10	PWM
PB3	D11	PWM
PB4	D12	
PB5	D13	
PC0	A0	Analog Input 0
PC1	A1	Analog Input 1
PC2	A2	Analog Input 2
PC3	A3	Analog Input 3
PC4	A4	Analog Input 4
PC5	A5	Analog Input 5
ADC6	A6 only on SMD	Analog Input 6
ADC7	A7 only on SMD	Analog Input 7
AREF	REF	Analog Reference
PC6/RESET	RST	Reset Pin
-	VIN	External Supply Voltage
VCC	5V	MCU Power
GND	GND	Ground

FTDI header pins

GND	Connected to ground of board
CTS	Clear to send signal needed when using the u-nex as an FTDI breakout board.
5V	Connected to 5V line/MCU
TX	Transmit data pin out
RX	Received data pin in
DTR	Data terminal ready, used to reset the MCU when programming using the arduino IDE. Also needed when using the u-nex as an FTDI breakout board.

Installing FTDI drivers:

The u-nex has an on-board USB to serial converter called the FTDI chip. When connected to the PC, the chip will enumerate (create) a COM port on your PC. This is the COM port you will use to communicate with the serial port of the u-nex.

The FTDI chip requires drivers for proper operation. You will need to download and install these prior to connecting the u-nex to your computer.

You can find the drivers available using the following URL:
http://www.ftdichip.com/Drivers/VCP.htm

Depending on your operating system, you can download the appropriate drivers. Always use the latest version available. Follow the installation procedure (usually double click the application to launch it and follow onscreen instructions). In case it is a compressed file, decompress the same to a folder and keep a note of the folder location and proceed with the following.

Once the installation is completed, follow the steps mentioned below to obtain your COM port.
1. Plug in a USB cable to the u-nex and computer.
2. Your operating system will pop up a message stating that a new device has been found and that the required drivers are being installed.
3. The operating system should automatically find and install the drivers on your system. In case of any errors, try selecting the browse option and locate the folder containing the files you downloaded from the FTDI site, it will use the files and install the appropriate drivers.
4. Once complete there will be a new device located under Ports (COM & LPT) which you can find in device manager. That is the COM port which represents the u-nex. Keep a note of the same.

Making sure the board works fine:

The u-nex is shipped with the arduino bootloader programmed to the chip and it has been tested by programming the LED blink sketch using the arduino IDE. We shall now perform a simple test to ensure the board is functional upon receipt. It is recommended to perform this test as soon as you receive the u-nex.

Note: The u-nex jumper is set appropriately depending on the version you have purchased. The lite version does not have the FTDI chip but can be still powered using the USB mini connector.

Please follow the steps mentioned below.
1. Remove the u-nex from the protective bag.
2. Gently connect a USB mini cable to the USB socket present on the board.
3. Connect the other end of the USB cable to a computer.
4. The u-nex will power up, this is indicated by the RED LED powering up.
5. Keep a note of the LD LED, it will initially blink fast on account of the arduino bootloader, after which it will blink at a steady rate - flashing on/off at an interval of 1second.
6. If the LED blinks as expected, the u-nex works as expected. It is recommended to disconnect the USB cable, keep the u-nex aside and continue reading this guide before proceeding further.

Powering the board:

As already demonstrated above, the u-nex can be powered using a USB mini cable, connected to the computer. The u-nex also has a VIN pin, to which an external supply voltage ranging from 7 to 12 V can be applied. The onboard 5V voltage regulator will convert the input voltage to 5V and supply it to the pin marked 5V. This can be used when interfacing external components; however the maximum current that can be delivered by the board is only 0.8A.

Note: Please ensure correct polarity when applying external voltage to the board. Not doing so *will* potentially damage the voltage regulator/MCU. When using the external voltage source, connect the positive pin to VIN and the ground pin to GND.

Using Arduino:

Configuring Arduino to use the u-nex - Windows

Arduino does not list the u-nex as one of the devices since we have not paid any royalty fees for the same. These are the things that enable us to keep the price of the u-nex very low. So you will have to copy a file to the appropriate folder in order for the u-nex to be listed. Follow the steps mentioned below:

1. Download and install the latest Arduino IDE. You can find the link below:
   http://arduino.cc/en/main/software
2. Once installed, navigate to the working directory of arduino. Examples for windows are shown below:
   C:\Program Files (x86)\Arduino\hardware\arduino
3. Download the "boards.txt" file from our repository.

4. Close the Arduino IDE if it is open. Copy the board.txt (from the u-nex repository) file into this folder, overwriting the existing file. You may wish to keep a backup copy of the old version of boards.txt, however there is no need for this. Launch the Ardunio IDE.
5. Navigate to the 'Tools' menu and select the option 'U-nex w/ ATmega328p' under 'Board'.
6. Navigate to the 'Tools' menu and select the COM port corresponding to the u-nex, which you had noted down earlier in this guide.
7. This is all that is needed to be done. We shall program the u-nex using arduino in the next section.

Configuring Arduino to use the u-nex – Mac OSX (by CJAndrade)

1. Download and install the latest Arduino IDE. You can find the link below:
   http://arduino.cc/en/main/software
2. Find the 'Arduino' application in Finder.
3. Right click on the App
4. Select 'Show Package Contents'
5. Navigate to Contents/Resources/Java/hardware/Arduino
6. Close the Arduino IDE if it is open. Replace the existing board.txt or rename the old one an add the new
7. Launch the Arduino IDE and navigate to the 'Tools' menu and select the option 'U-nex w/ ATmega328p' under 'Board'.
8. Navigate to the 'Tools' menu and select the COM port corresponding to the u-nex, which you had noted down earlier in this guide.
9. This is all that is needed to be done. We shall program the u-nex using arduino in the next section.

Programming using Arduino – Making sure the board works – Part 2

The arduino bootloader present on the board enables you to program the u-nex using the arduino IDE, connected using the USB cable. Follow the steps to program the u-nex using the ARduino IDE.

1. Connect the u-nex to the computer using a USB cable. Ensure the COM port is enumerated.
2. Make sure the correct board and serial port is selected under the tools menu.
3. Navigate to the File menu and select the Blink sketch by navigating as follows:
   a. File -> Examples -> 01. Basics -> Blink
4. A new IDE window will open. Select the upload button within that IDE. The sketch will automatically compile and will be uploaded to the u-nex. You will see the confirmation as illustrated below. When the code is being uploaded, you will see the Tx and Rx LEDs flash rapidly, indicating that serial communication is active.
5. Once the code is uploaded, the board will automatically reset and the new sketch will start running.
6. Congratulations! You have successfully programmed the board using the Arduino IDE. Play around with the sketch by changing the delay between LED being turned On/Off, upload the new sketch to the board and watch it come alive.

Programming using ICSP

The u-nex can also be programmed using a traditional ICSP device. You may want to this this in case you want to re-write the arduino bootloader or erase it to use the u-nex as a standard AVR development board.

Due to the compact nature of the board, we were unable to house a standard ICSP header, but you can use the shield which is available for purchase separately. Apart from that you can also hook-up wires to the pins needed to program the u-nex using an ICSP.

The following illustration shows the pins and the legend of the same. All the pins needed are contained on the right hand section of the board to enable easier programming.

Use the standard software you normally use to program the chip example AVR dude.

Configuring the solder jumpers:

The u-nex contains two solder jumpers on the bottom section of the board. These jumpers enable you to connect the serial port lines (Tx and Rx) of the atmega328p to suit your requirements.

You can essentially connect the serial lines directly to the FTDI chip present on board (solder jumper connected to MCU), or to the FTDI header pins (solder jumper connected to HDR). Alternately you can isolate them from both if your application demands it.

Programming using an external FTDI cable or u-nex

The u-nex can also be programmed using an external FTDI cable or u-nex. For this to work you would need to have the arduino bootloader present on the board. The board ships directly with it.

Note: It is recommended not to power the board by any other means when using the FTDI cable.

Simply configure the solder jumpers to connect the ATmega328p jumper to the external header pins and connect the FTDI cable to the header with proper orientation.

If you have another u-nex with FTDI chip, you can configure the jumpers to attach the FTDI chip output to the header pin, thus making it a FTDI breakout board.

Once done, you can connect the lines directly from one u-nex to another, the pins are position to go in directly as illustrated below.

Once you have the setup mentioned above, simply configure the arduino IDE to use the correct serial port and upload the sketch to the board. That's all it takes.

Document Revision:

Version	Date	Comments
1.0	11th May 2014	Initial Release
1.1	1st August 2014	Added support for 'board.txt' update in Mac OSX