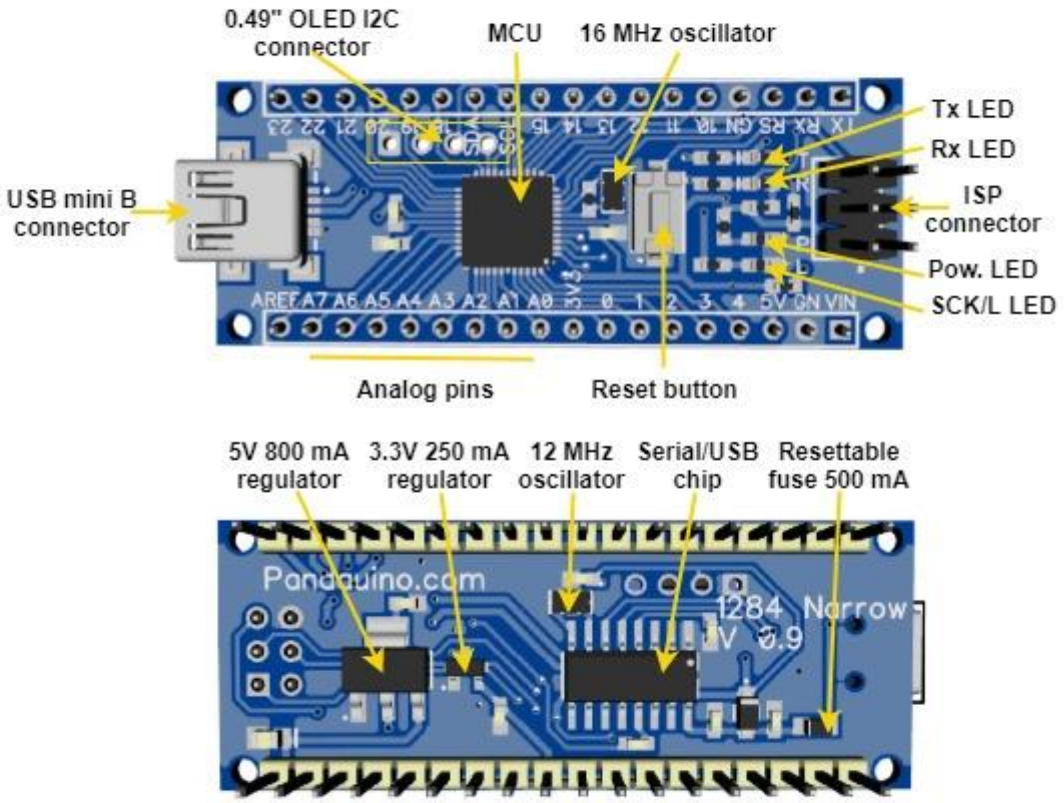
644/1284 Narrow
User's manual

Presentation of the board

644 Narrow and 1284 Narrow were inspired by the Arduino Nano board. Their functionality is quite the same but **instead of featuring an Atmega328 MCU they implement an Atmega644 or Atmega1284.**

These boards are as small as possible, to fit on a breadboard, given that they feature an MCU, a USB plug, a USB/Serial converter, an ISP port, an I2C port, a power circuit, a couple of LEDs and a reset button. In addition to a usual layout you will find on the board a **special I2C port** (GND-5V-SCL-SDA) where you can solder a 0.49” OLED display.

644/1284 Narrow Global view



As compared to the Nano and Uno board they feature:

- 4KB/16KB instead of 2 KB of RAM
- 64KB/128KB instead of 32 KB of Flash memory
- 2KB/4KB instead of 1 KB of EEPROM
- One more USART port (Atmega1284)
- 12 more digital I/O

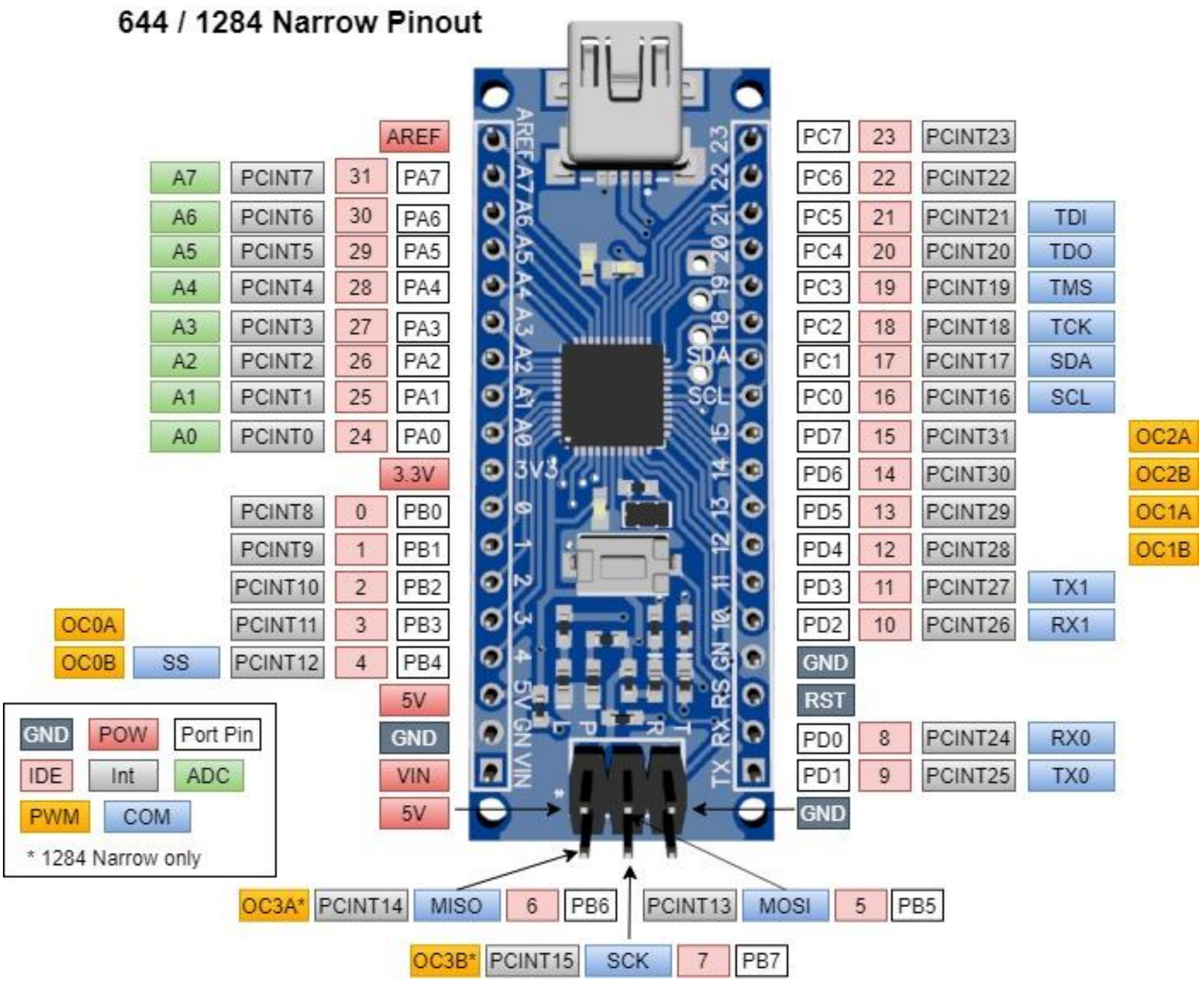
Even though the line of product of Atmega644/Atmega1284 is different, they remain part of the latest 8 bits MCU from Atmel with quite the same architecture as that of the Atmega328. So if you are familiar with the Atmega328 based boards (Uno, Nano etc …) you will not have any problem using the Narrow boards. Programming in the Arduino environment is the same.

You can download the Atmega644/1284 data sheet from this link

Board specifications

	644 Narrow	1284 Narrow
MCU	Atmega644pa-mu	Atmega1284p-mu
Operating voltage	Vin: 6.2 to 15 volts USB: 4.75 to 5.25 volts On board 5V (800 mA) and 3.3V (250 mA) regulators.	Vin: 6.2 to 15 volts USB: 4.75 to 5.25 volts On board 5V (800 mA) and 3.3V (250 mA) regulators.
Max current	USB: 500 mA limited by a PTC. Vin: 800 mA @ 6.2V, 25°c	USB: 500 mA limited by a PTC. Vin: 800 mA @ 6.2V, 25°c
Flash	64 Kb	128 Kb
RAM	4 Kb	16 Kb
EEPROM	2 Kb	4 Kb
Digital I/Os	24	24
Analog I/Os	8	8
PWM	6	8
SPI	1	1
USART	1	2
I2C	1	1
USB	1	1
Size	1097 mm^2	1097 mm^2
Weight	8 g	8 g

Pinout


644 / 1284 Narrow Pinout
AREF
A7 PCINT7 31 PA7
A6 PCINT6 30 PA6
A5 PCINT5 29 PA5
A4 PCINT4 28 PA4
A3 PCINT3 27 PA3
A2 PCINT2 26 PA2
A1 PCINT1 25 PA1
A0 PCINT0 24 PA0
3.3V
PCINT8 0 PB0
PCINT9 1 PB1
PCINT10 2 PB2
OC0A PCINT11 3 PB3
OC0B SS PCINT12 4 PB4
5V
GND
VIN
5V
PC7 23 PCINT23
PC6 22 PCINT22
PC5 21 PCINT21 TDI
PC4 20 PCINT20 TDO
PC3 19 PCINT19 TMS
PC2 18 PCINT18 TCK
PC1 17 PCINT17 SDA
PC0 16 PCINT16 SCL
PD7 15 PCINT31 OC2A
PD6 14 PCINT30 OC2B
PD5 13 PCINT29 OC1A
PD4 12 PCINT28 OC1B
PD3 11 PCINT27 TX1
PD2 10 PCINT26 RX1
GND
RST
PD0 8 PCINT24 RX0
PD1 9 PCINT25 TX0
GND
OC3A* PCINT14 MISO 6 PB6
PCINT13 MOSI 5 PB5
OC3B* PCINT15 SCK 7 PB7
GND POW Port Pin
IDE Int ADC
PWM COM
* 1284 Narrow only
AREF A7 A6 A5 A4 A3 A2 A1 A0 3V3 0 1 2 3 4 5V GN VIN
23 22 21 20 19 18 SDA SCL 15 14 13 12 11 10 GN RS RX TX
L P R T

Installation

Arduino programming environment

Of course, if you intend to develop your program in the Arduino Environment you must have the Arduino IDE installed. You can download it from https://www.arduino.cc/en/main/software

The following installation procedure was tested with the **Arduino IDE version 1.8.13**

Automatic install under Windows

Windows users can simplify the installation process:

1. **Install the MightyCore**. Follow the installation process: **https://github.com/MCUdude/MightyCore#how-to-install.**

2. **Download and execute** https://github.com/mrguen/MightyCore/blob/master/MightyCore-Pandauino-WINDOWS-installer.exe

The installer is not signed so it is normal if you get a warning.

It will:

- Install the boards definitions in your local sketch/hardware folder
- Install the CH340 driver (close the driver installer if it is already installed)

You can skip the next explanations on “CH340 driver” installation and “Board definition”, but if you need to troubleshoot the automatic installation, and have a look at the Arduino IDE configuration.

CH340 driver

The USB/Serial adapter is the well know CH340G chip. It is a Chinese chip that is implemented on most Nano clones and works well.

You will need to install its driver from http://www.wch-ic.com/downloads/CH341SER_EXE.html

You can also refer to https://sparks.gogo.co.nz/ch340.html

Once the driver is installed and you connect the board to your computer using a USB plug you should see it listed in the USB peripherals with a name like “CH340”

Board definition / MightyCore

The Arduino IDE does not include the Atmega644 and Atmega1284 support by default. You can find a couple of boards definitions on GitHub see

https://github.com/search?q=Atmega644
https://github.com/search?q=Atmega1284

But you will probably like the MightyCore. There are two possibilities to install the MightyCore boards definition including Narrow boards:

- Copy the files from the repository https://github.com/mrguen/MightyCore into your personal sketchbook /hardware folder. This is the preferred option to have a coherent configuration.

- Or modify the automatic install of the MightyCore from the Arduino. It is a possibility if you already have the MightyCore installed but it might fail in future versions of the MightyCore.

OPTION A – INSTALL BOARDS DEFINITION IN YOUR SKETCHBOOK DIRECTORY

- Go to https://github.com/mrguen/MightyCore
- Click the green button Code – Download zip and save the archive to your computer
- Unzip its content in your sketchbook /hardware folder.
  For example, on Windows 10 standard install it is located in
  C:\Users\USER_NAME\Documents\Arduino\hardware
  Create the \hardware folder if necessary.

OPTION B – MODIFY THE MIGHTYCORE AUTOMATIC INSTALL

This procedure is tested for Arduino IDE 1.8.13 and MightyCore 2.1.1.

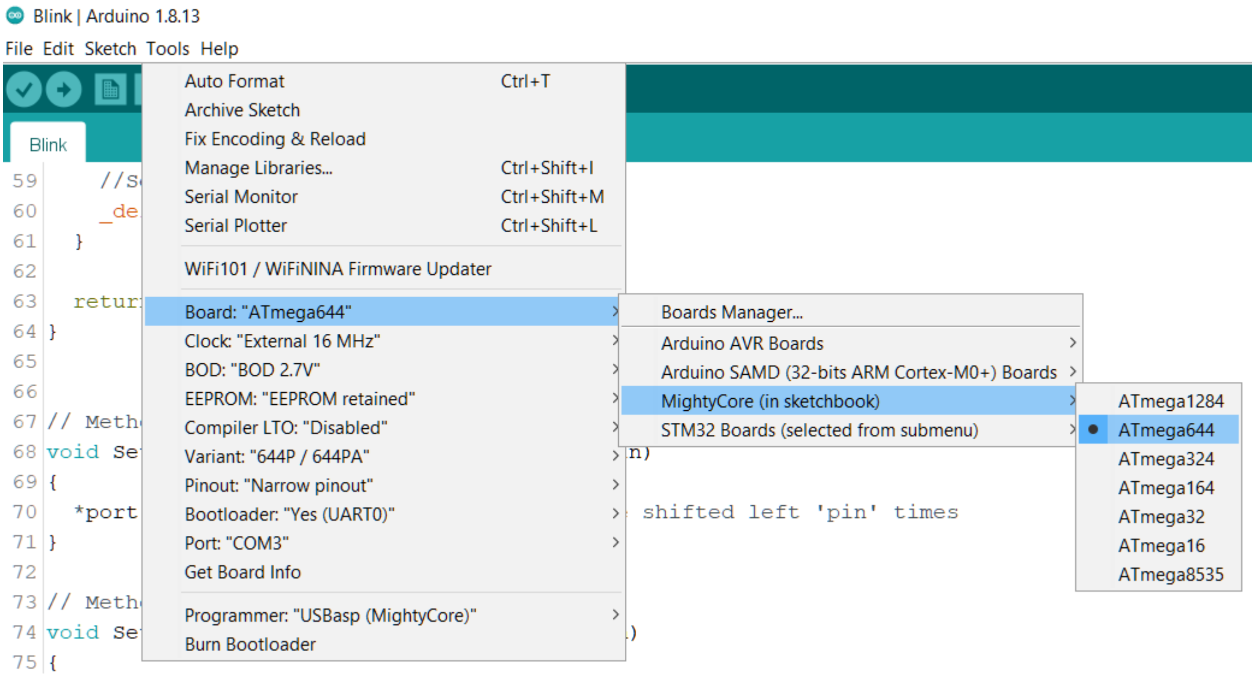
- Install the MightyCore. Follow the installation process: **https://github.com/MCUdude/MightyCore#how-to-install.**

- Download the variant file from https://github.com/mrguen/MightyCore/tree/master/avr/variants/narrow and place it inside your MightyCore install directory. For example on Windows 10 standard install it is located in C:\Users\ USER_NAME \AppData\Local\Arduino15\packages\MightyCore\ hardware\avr\VERSION\variants

- Modify the MightyCore install board.txt (in C:\Users\USER_NAME \AppData\Local\Arduino15\packages\MightyCore\ hardware\avr\VERSION) by adding these lines where appropriate:

```
1284.menu.pinout.narrow=Narrow pinout
1284.menu.pinout.narrow.build.variant=narrow
1284.menu.pinout.narrow.build.bootloader_led=B7
644.menu.pinout.narrow=Narrow pinout
644.menu.pinout.narrow.build.variant=narrow
644.menu.pinout.narrow.build.bootloader_led=B7
```

Arduino IDE configuration when using the MightyCore


Blink | Arduino 1.8.13
File Edit Sketch Tools Help
Blink
Boards Manager...
Arduino AVR Boards
Arduino SAMD (32-bits ARM Cortex-M0+) Boards
MightyCore (in sketchbook)
STM32 Boards (selected from submenu)
ATmega1284
● ATmega644
ATmega324
ATmega164
ATmega32
ATmega16
ATmega8535

```
    //S
    _de
  }

  retur
}


// Meth
void Se                                      n)
{
  *port                                      shifted left 'pin' times
}

// Meth
void Se                                      )
{
```

Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "ATmega644"
Clock: "External 16 MHz"
BOD: "BOD 2.7V"
EEPROM: "EEPROM retained"
Compiler LTO: "Disabled"
Variant: "644P / 644PA"
Pinout: "Narrow pinout"
Bootloader: "Yes (UART0)"
Port: "COM3"
Get Board Info
Programmer: "USBasp (MightyCore)"
Burn Bootloader

In the Arduino IDE **Tools** menu, you should select these options:

Board: MightyCore (in sketchbook)/ Atmega644 or Atmega1284

Clock: External 16 MHz

BOD: BOD 2.7V

EERPOM: “EEPROM retained”, or not depending on your application

Compiler LTO: disabled but you can also use LTO enabled

Variant: P or PA variant

Pinout: Narrow pinout

Bootloader: Yes (UART0)

Port: the COM port given by your system (once the bootloader has been installed and the board connected through the USB port)

Programming

You can program the board directly through the USB connector, since the bootloader is already loaded. But you can also program it through the ISP socket. In this case it will write over the bootloader.

SPI programming

You can program the board using a programmer like the USBAsp or any of the many other supported programmers.

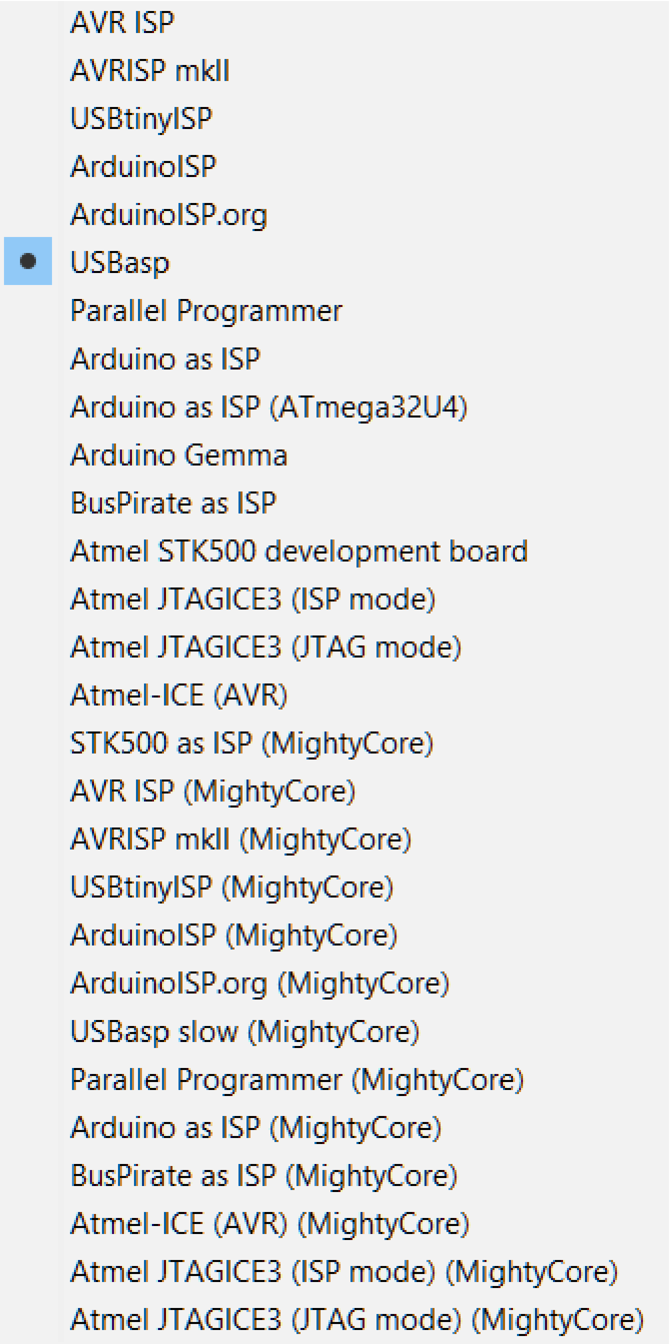
1. Install your programmer on your system
2. Connect the programmer to the ISP port (2*3 header) on the device and to the computer. It should be properly listed as a device by your system. **Make sure you connect in the right direction with the pin number 1 identified by ***
3. Select the proper programmer in **Tools/Programmer**
4. Select the menu **Sketch/Upload Using Programmer**.

USB programming

When the bootloader is installed, you can also program the device through a USB port.

The bootloader has been uploaded on the device already but it might have been written over if you programmed the board using the ISP socket. In this case, burn the bootloader by uploading it to the board with a programmer (see steps 1-3 of SPI programming): use the menu **Tools/Burn Bootloader.**

To upload your code you can simply **press the arrow button** in the main window of the Arduino IDE or the menu **Sketch/Upload**


AVR ISP
AVRISP mkII
USBtinyISP
ArduinoISP
ArduinoISP.org
● USBasp
Parallel Programmer
Arduino as ISP
Arduino as ISP (ATmega32U4)
Arduino Gemma
BusPirate as ISP
Atmel STK500 development board
Atmel JTAGICE3 (ISP mode)
Atmel JTAGICE3 (JTAG mode)
Atmel-ICE (AVR)
STK500 as ISP (MightyCore)
AVR ISP (MightyCore)
AVRISP mkII (MightyCore)
USBtinyISP (MightyCore)
ArduinoISP (MightyCore)
ArduinoISP.org (MightyCore)
USBasp slow (MightyCore)
Parallel Programmer (MightyCore)
Arduino as ISP (MightyCore)
BusPirate as ISP (MightyCore)
Atmel-ICE (AVR) (MightyCore)
Atmel JTAGICE3 (ISP mode) (MightyCore)
Atmel JTAGICE3 (JTAG mode) (MightyCore)

Programming examples

Blink

Usually you might want to check that everything is working properly by programming the sketch Blink.ino that you will find in the Arduino IDE menu **Files/Examples/Basic/Blink**

The led labelled “L” on the board will be blinking at low speed.

If id does not, check that you have defined the board as explained in Board definition / MigthyCore

2nd Serial port

There are two serial ports.

1) You can test that they are working using the sketch that you will find in the Arduino IDE menu **File/Examples/Communication/MultiSerial**

This sketch needs to have a serial device attached to the Serial 1 port (pin 10 RX1, pin 11 TX1).

2) If you don’t have such a device you can simply try this sketch

https://github.com/mrguen/644-1284-Narrow/tree/master/examples/Endless_Serial_0_1/Endless_Serial_0_1.ino/

Connect pins TX and 10 together.

Program the sketch through the USB port. In the Serial Monitor, if you type a letter to send to the board through the Serial port, it will display endlessly on the screen because it is sent back by serial “0” Tx pin to the serial monitor and also to serial “1” Rx, that writes again to serial “0” Tx.

0.49” OLED display

There is a group of four plated through-holes on the board to solder a tiny 0.49” OLED 64x32 pixels display. Not all I2C OLED modules connections are arranged in the same order so check it before soldering in your display. These holes are connected to

GND – 5V – SCL - SDA

Attention: the labels SDA and SCL on the board are for the pins on the edge of the board and not for this connector that has SCL and SDA arranged the other way and that are not labelled.

The OLED module is provided apart because you might want to use it remotely with Dupont cables to a 4 pins header that you would solder on the board.

There are two programming examples available.

Using the Adafruit library

A generic library, wildly used.

In the downloaded narrow repository **/examples/test_SSD1306_Adafruit**
or from
https://github.com/mrguen/644-1284-Narrow/tree/master/examples/test_SSD1306_Adafruit

The Adafruit library is mostly interesting is you want to display shapes, or even bitmaps. See https://learn.adafruit.com/adafruit-gfx-graphics-library

Using the Greiman library

A light library to print nice text.

In the downloaded narrow repository **/examples/test_SSD1306_Ascii**
Or from
https://github.com/mrguen/644-1284-Narrow/tree/master/examples/test_SSD1306_Ascii

Initially based on https://github.com/greiman/SSD1306Ascii library, it is mostly limited to text rendering but with a large and nice choice of fonts.

1024 samples FFT

You can use the 16 KB of RAM of the 1284 Narrow to compute a 1024 samples FFT.

See the downloaded narrow repository **/examples/Test_FFT_ADC_1284_Narrow**
Of
https://github.com/mrguen/644-1284-Narrow/tree/master/examples/Graphicstest_ST7789_Fast_Narrow_TFT_240x240_balloon

A tutorial is available at https://www.instructables.com/1024-Samples-FFT-Spectrum-Using-an-Atmega1284/

Schematics

See the pictures in the downloaded repository

/doc/Narrow v1.0 schematics – 1.jpg
/doc/Narrow v1.0 schematics - 2.jpg

Or at
https://github.com/mrguen/644-1284-Narrow/blob/master/doc/Narrow%20v1.0%20schematics%20-%201.jpg

https://github.com/mrguen/644-1284-Narrow/blob/master/doc/Narrow%20v1.0%20schematics%20-%202.jpg

Troubleshooting

Problem	**Possible solution**
The power led (blue) does not light up.	Check the voltage at the 5V pin. It should be between 4.5V and 5.25V If not, maybe the polyswitch "PS" or the diode D101 or the regulator U2 might be damaged. But it is unlikely since the polyswitch should protect the circuit. Most common cause is a bad connection to the voltage source (Vin or USB).
After you burn the bootloader the orange led does not blink twice	This led is supposed to constantly blink twice when the bootloader is present and no program was downloaded. If not, the led pin definition might be wrong. Check the proper configuration of the environment. Board definition / MigthyCore
When trying to download from the Arduino IDE you get a series of errors: avrdude: stk500_recv(): programmer is not responding avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x1e …	Check that you installed the CH340 driver Download the bootloader. See Programming / USB programming If everything is properly configured when you connect the board using an USB cable it should be enumerated as a COM port "USB- SERIAL CH340". If not, check the USB cable and connections.