# Installing the Odin Console Software

V1.0

Laurence Barker G8NJJ March 2018

This guide describes how to download, install and load the Arduino software for the Odin console. The guide assumes that you are using the Arduino Integrated Development Environment (IDE) running on a windows platform. For users with different operating systems, different folder locations will probably apply.

# Install the Arduino IDE

The Arduino IDE is downloaded from the Arduino web page. The download links are on this page:

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

Download and install the IDE. When you run it for the first time, it will look something like:

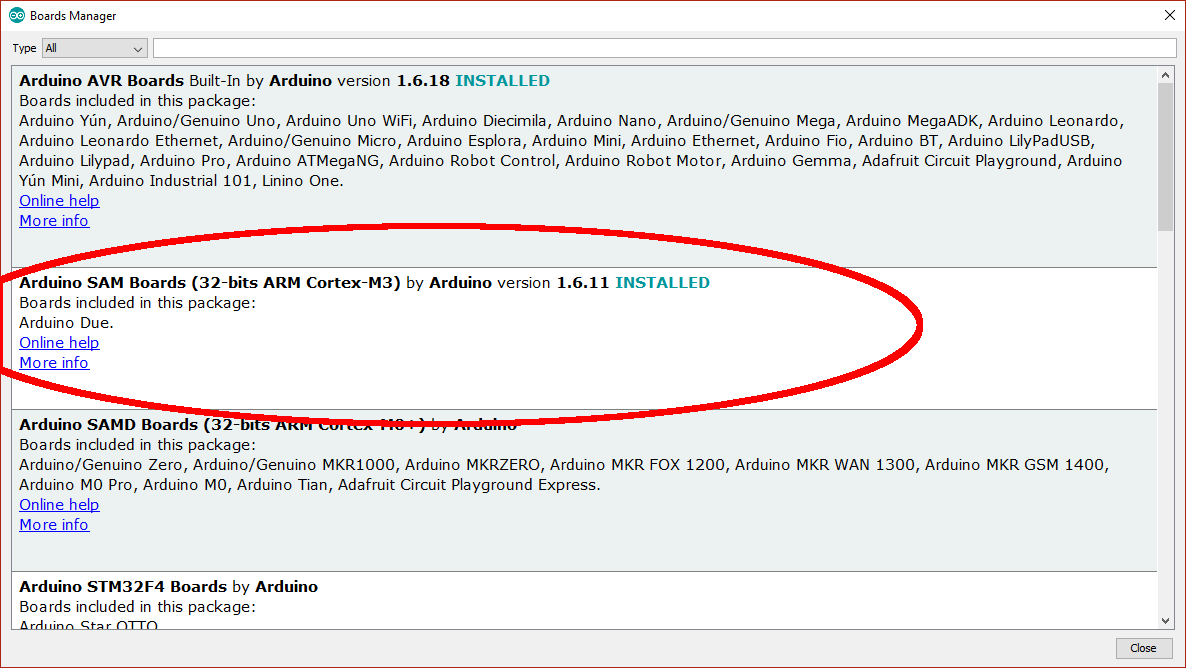


This is showing you a new, blank program. Arduino programs are called “sketches”.

# Add Support for the Due Board

As shipped the Arduino IDE can build code for some of the processor types used in the Arduino range, but not for the Arduino “Due” used in this project. A simple download will add the Due:

1. Open the Arduino IDE
2. Click “Tools|Board|Boards manager” on the menu
3. Scroll down to the entry for “Arduino SAM Boards (32-bit ARM Cortex-M3)” and click “install”
4. Your screen should now look something like this:



# Install Libraries into the Arduino IDE

The next step is to install 4 libraries into the Arduino library. This will provide access to the code that we have used as part of the Odin build.

The Arduino system loads libraries into a folder it created on your computer; usually that folder is installed into the “documents” folder called “Arduino\libraries”. On my computer that folder is “C:\Users\loz barker\Documents\Arduino\libraries”. Use windows explorer to find that folder so you know where it is.

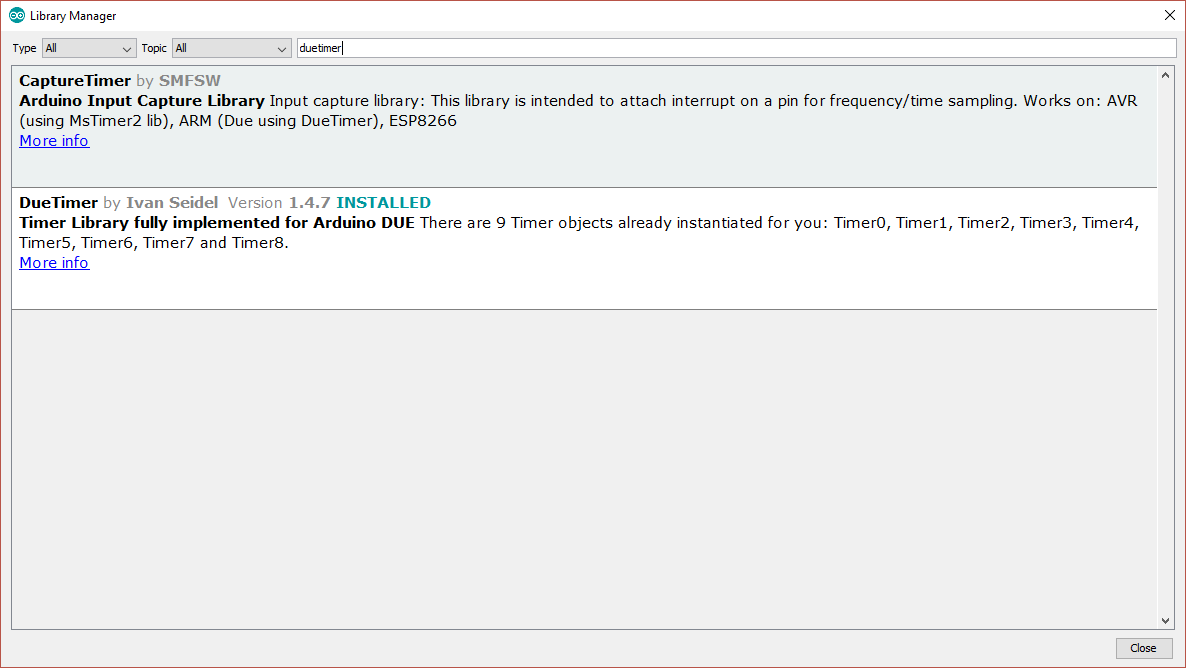
The 4 required libraries are:

|  |  |
| --- | --- |
| DueFlashStorage | Provides persistent storage so settings are kept while power is removed |
| DueTimer | Provides access to the Arduino’s timer, so the code can synchronise to a 1 millisecond time count |
| Encoder | Controls the optical VFO encoder |
| ITEADLIB\_Arduino\_Nextion | Controls the touchscreen display |

The first two are accessed through the the Arduino library manager; the second two have to be installed manually.

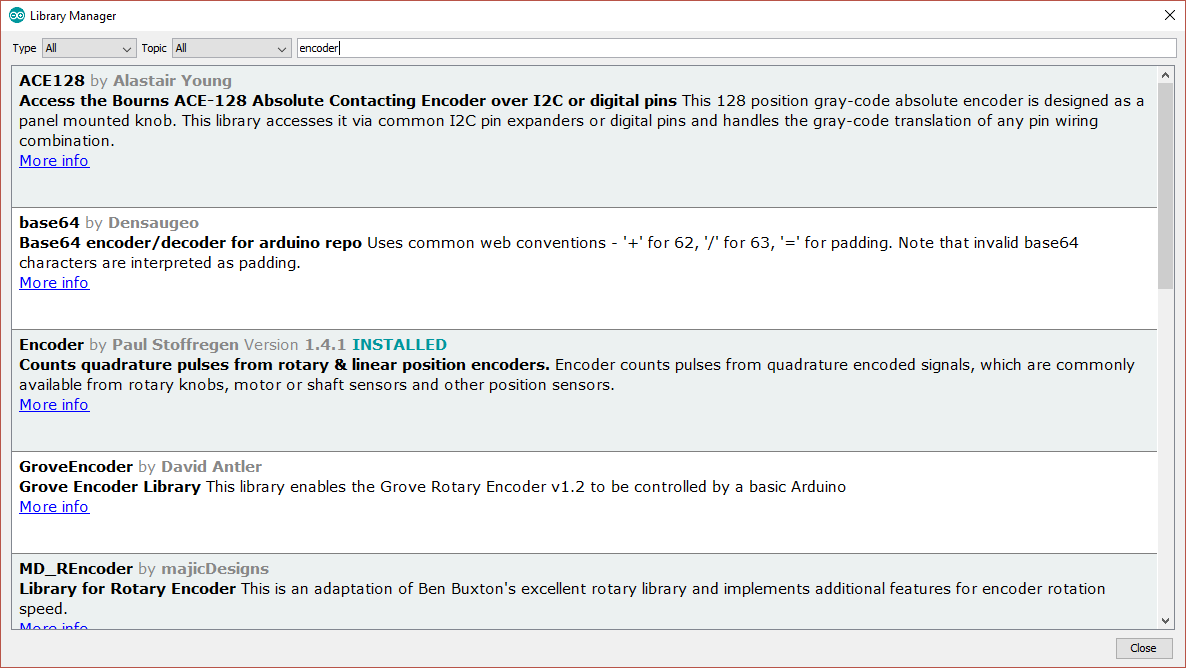
## DueTimer

1. Open the Arduino IDE
2. Click "Sketch | Include Library | Manage libraries..." on the menu
3. In the library manager type "due timer" where it says "filter your search" and hit enter
4. Find "DueTimer" by Ivan Seidel and click "install"
5. You should now have a folder “Documents\Arduino\libraries\DueTimer”
6. Your screen will look something like this:



## Encoder

1. In the library manager type "encoder" where it sayds "filter your search" and hit enter
2. Find "encoder" by Paul Stoffregen and click "install"
3. Your screen will look something like this:



## DueFlashStorage

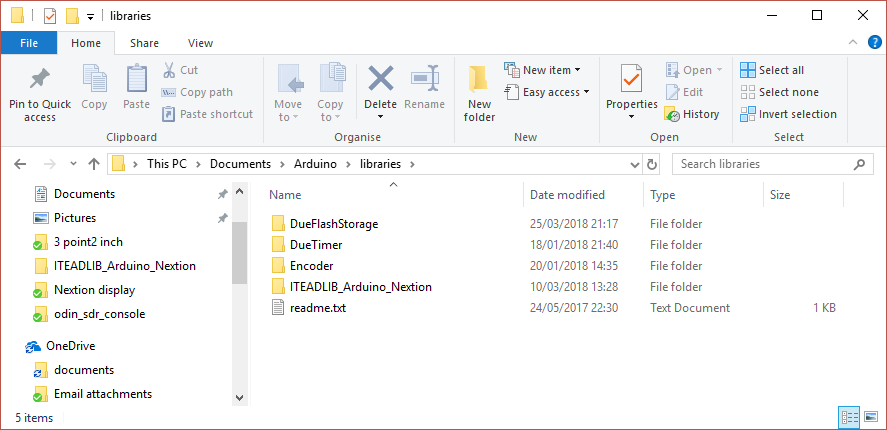
This needs to be installed manually. The procedure is as follows:

1. Visit the repository on github: <https://github.com/sebnil/DueFlashStorage>
2. Click “clone or download” then “download zip”
3. Store the zip file on your PC for example in the “downloads” folder
4. Open the zip file and extract all files. You will now have a folder “DueFlashStorage-master” which will hold one folder also called “DueFlashStorage-master”
5. Rename the second folder “DueFlashStorage” (remove the “-master” part)
6. Copy that whole folder to your “documents\arduino\libraries” folder

## ITEADLIB\_Arduino\_Nextion

This needs to be installed using a similar process:

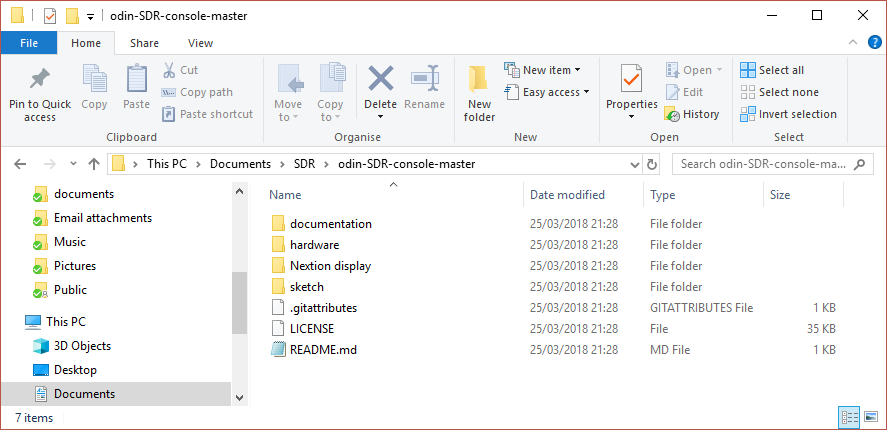
1. Visit the repository on github: <https://github.com/itead/ITEADLIB_Arduino_Nextion>
2. Click “clone or download” then “download zip”
3. Store the zip file on your PC for example in the “downloads” folder
4. Open the zip file and extract all files. You will now have a folder “ITEADLIB\_Arduino\_Nextion-master” which will hold one folder also called “ITEADLIB\_Arduino\_Nextion-master”
5. Rename the second folder “ITEADLIB\_Arduino\_Nextion” (remove the “-master” part)
6. Copy that whole folder to your “documents\arduino\libraries” folder
7. (This is the library published by the display manufacturer. Be aware there is some foul language in the "html" folder - delete the entire "html" folder if you do not want that)
8. Your “documents\arduino\libraries” folder should now look something like:



The ITEADLIB folder needs to be patched in the next phase!

# Download the Odin Software Repository

1. Visit the repository on github: <https://github.com/laurencebarker/odin-SDR-console>
2. Click “clone or download” then “download zip”
3. Store the zip file on your PC for example in the “downloads” folder
4. Open the zip file and extract to your PC; for example into a folder “SDR” in “documents”
5. There will be a folder called “odin-SDR-console-master” in your “SDR” folder and its contents will look something like:



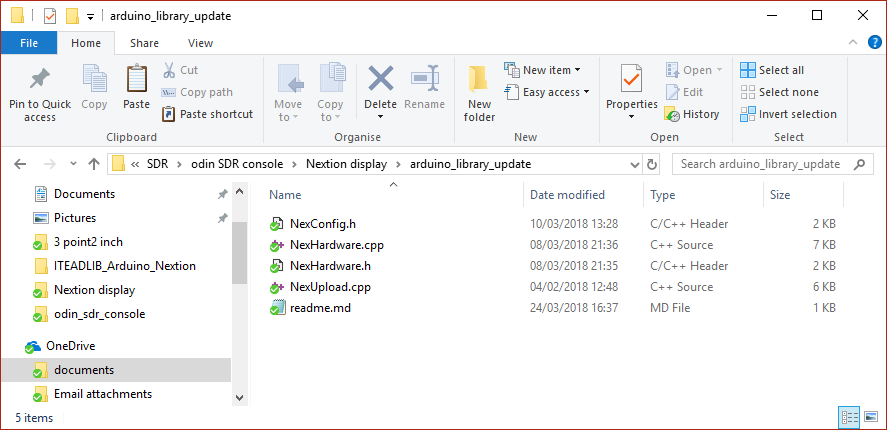
There are several folders:

|  |  |
| --- | --- |
| Documentation | The user guide and this installation guide |
| Hardware | The schematics and PCB layouts for the console PCB |
| Nextion Display | Files for 2 things:   * For the Nextion display itself, setting out the layouts of the screens used * Files to patch the Arduino library for the display |
| Sketch | The Arduino program for the console. |

## Patch the ITEADLIB Library

Four files (plus a readme file) need to be copied from the Odin repository to the ITEADLIB folder in the Arduino libraries.

1. Open the folder “nextion display\arduino\_library\_update”
2. It will have files as follows:

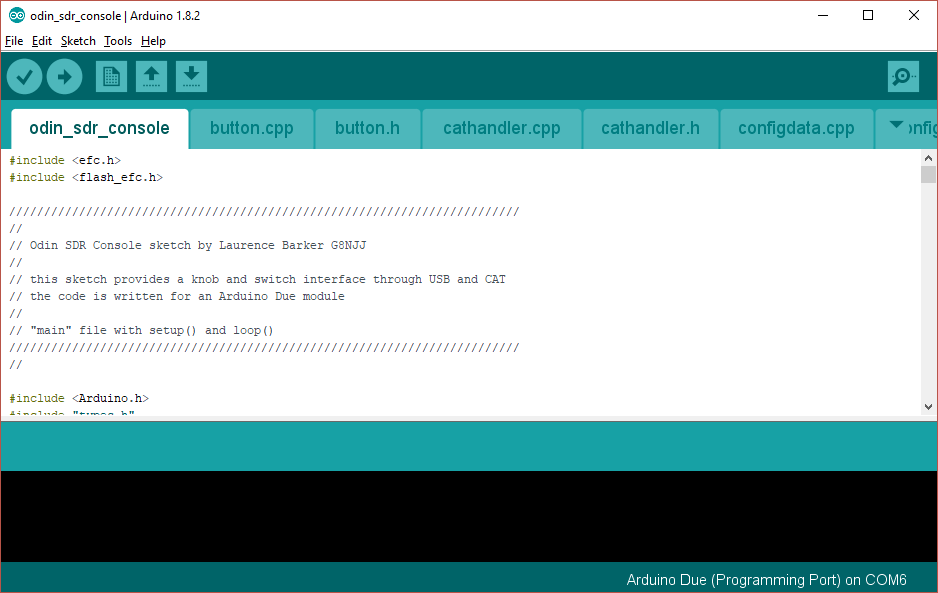


1. Select then copy those files
2. Navigate to your folder "documents\arduino\libraries\ITEADLIB\_Arduino\_Nextion"
3. Paste the 5 files there. 4 existing files will be replaced and the readme file will be added.

## Build the code

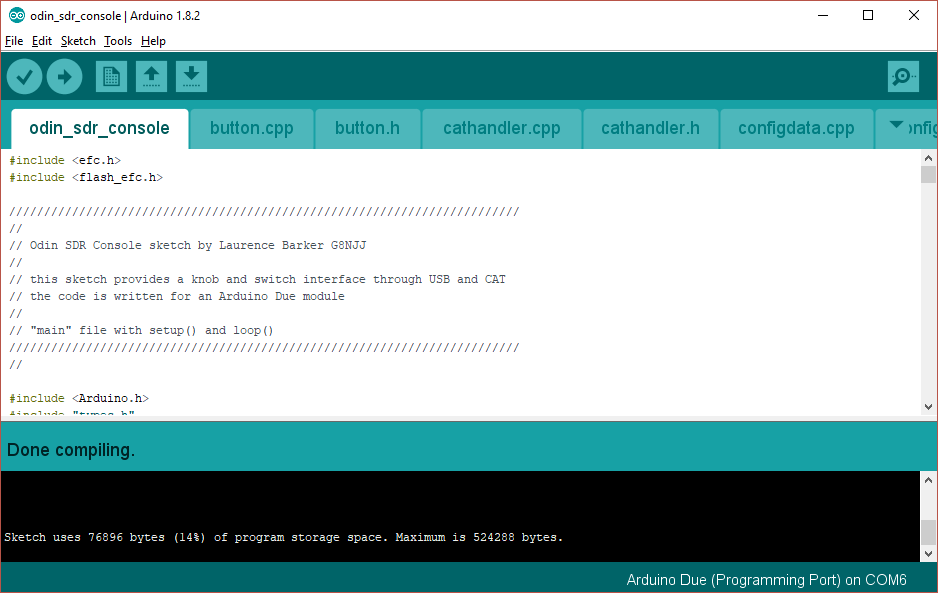
To open the Odin software sketch:

1. Run the Arduino IDE
2. Use the "File|Open..." menu command
3. Navigate to "odin\_sdr\_console.ino" in folder "Documents\SDR\odin-SDR-console-master\sketch\odin\_sdr\_console" and click "open"
4. you should now see the files listed in tabs above the editor window



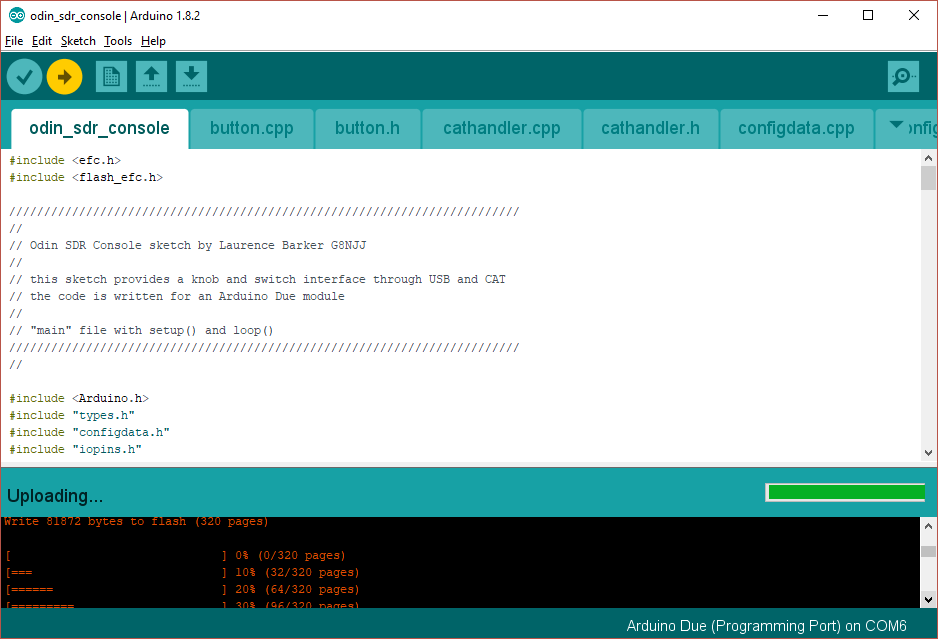
You now need to tell the IDE what kind of board it is compiling for, and which serial port to use to connect to it.

1. Connect a USB cable between the Arduino programming port (next to the black power connector) and your PC.
2. It may be necessary to install device drivers at this point – follow any instructions.
3. Click "board" on the "tools" menu and select "Arduino Due (programming Port) from the list"
4. Click “port” on the “tools” menu and choose the Arduino COM port listed (mine is COM6)
5. Click "Verify/compile" on the "sketch" menu to compile
6. (A message “compiling sketch…” will appear. This will take around a minute and should result in a message saying the % of program space used)



Finally you need to upload the code to your Arduino:

* Click "Upload" on the "sketch" menu to upload to the Arduino
* A simple progress bar will show in the bottom window of the IDE, twice - for each of "programming" and "verify"
* When it has successful finished the last message will be "CPU reset"





Your Arduino should now be executing the Odin code!