Manual:

EtherCATduino

Pieter-Jan Buntinx

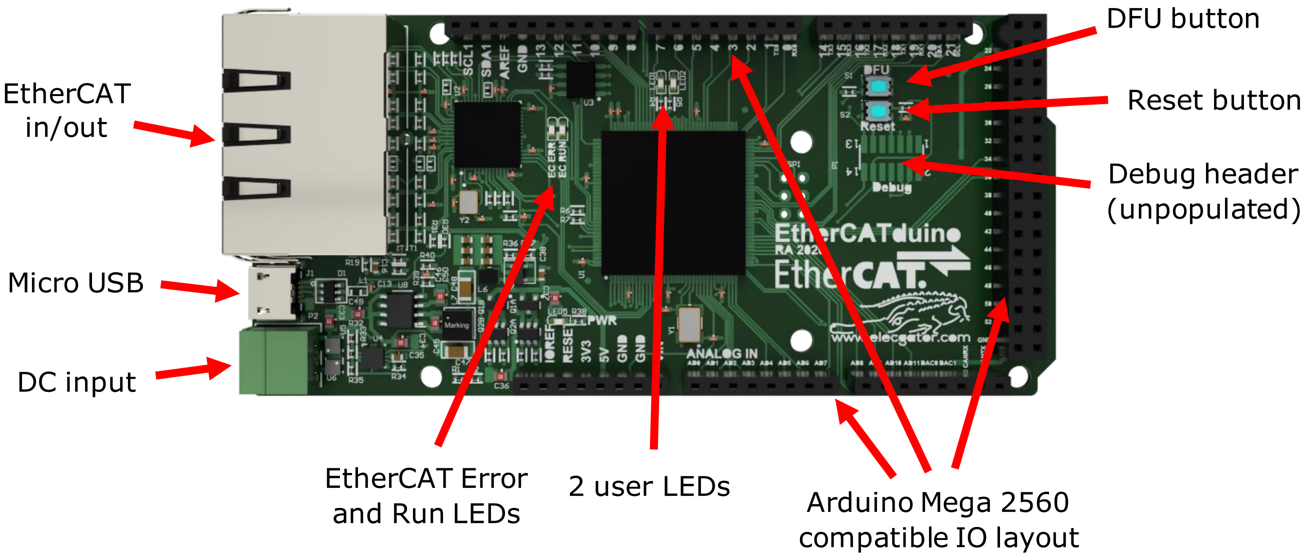
9-6-2021

www.elecgator.com

1 EtherCATduino development board

1.1 General

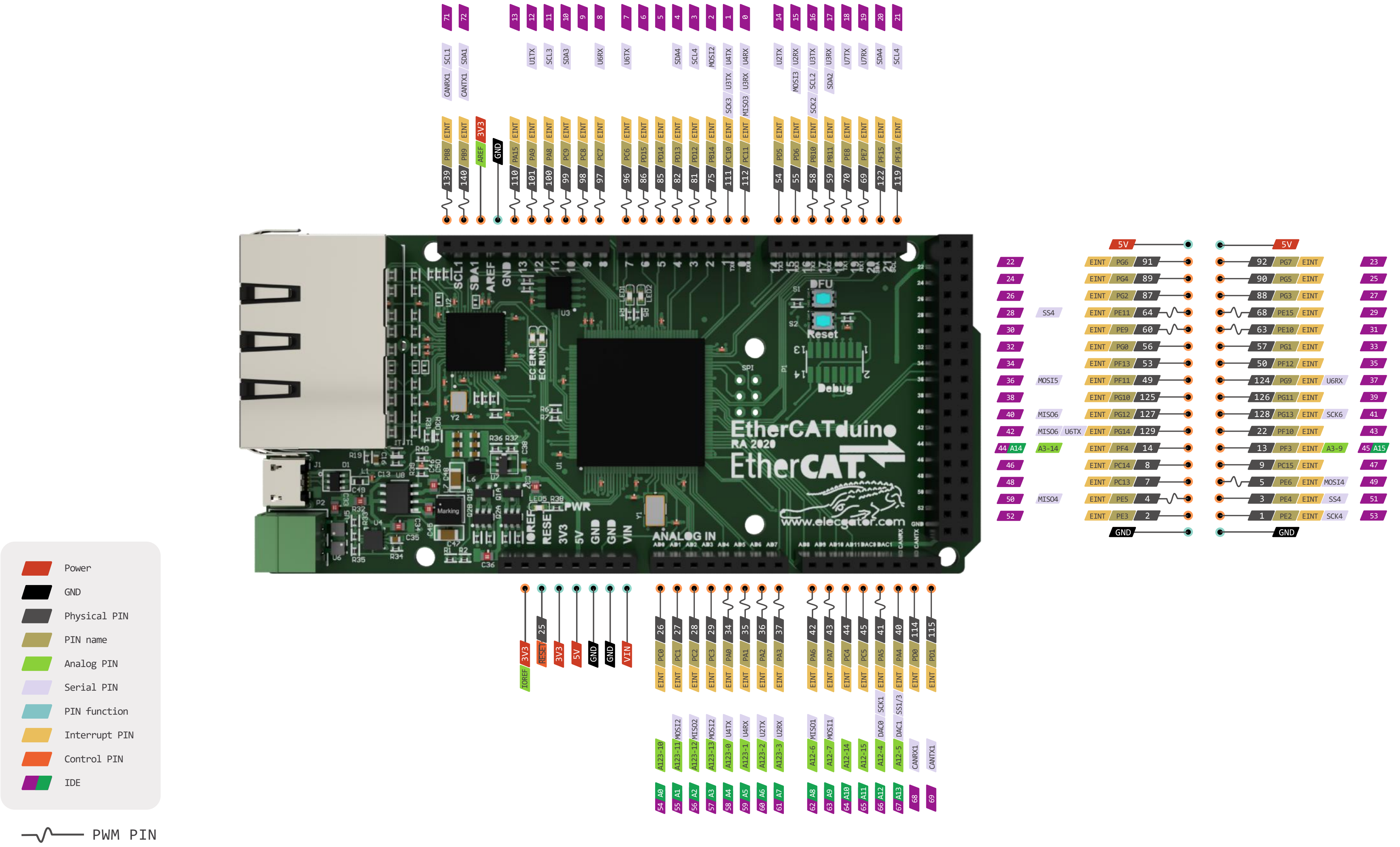
The EtherCATduino is an Arduino compatible development board based on an ARM-based STM32F74S6 microcontroller that uses a LAN9252 chip as a 2-port EtherCAT slave controller. This board has a built-in DC-DC that supports voltages up to 28 Volts, a micro USB port, 2 RJ45 connectors (for EtherCAT) and the same pinout and overall form factor as an Arduino Mega 2560.


EtherCAT in/out
Micro USB
DC input
EtherCAT Error and Run LEDs
2 user LEDs
Arduino Mega 2560 compatible IO layout
DFU button
Reset button
Debug header (unpopulated)

1.2 Specifications

Powerful ARM Cortex-M7 microcontroller at 216 MHz
Micro USB port for programming and power (max 500 mA)
DC input port supporting 7 to 28 Volts (Wurth connector: 691361300002)
54 general purpose input/outputs
12 analog inputs
2 DAC outputs
2 RJ45 connectors for EtherCAT in and out

1.3 Pin description

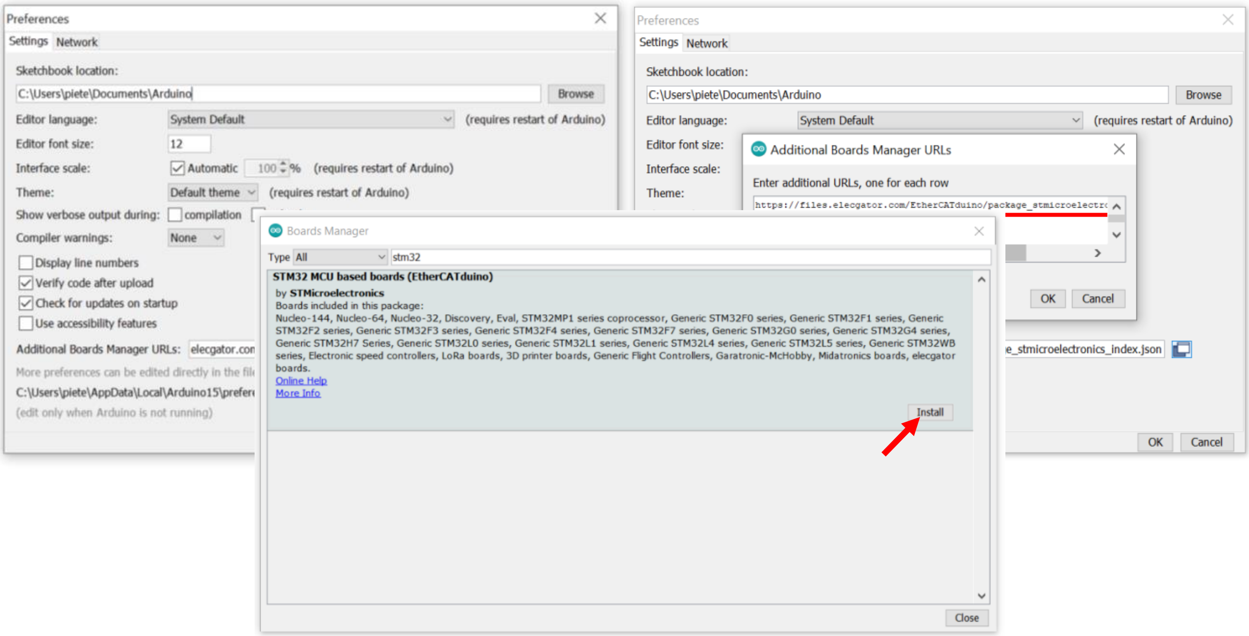


1.4 Arduino IDE

1.4.1 Installation

To have the EtherCATduino board available in the Arduino IDE use the following directions:

- In the Arduino IDE, navigate to: File > Preferences…
- Add the following URL to the "Additional Boards Manager URLs" dialog: https://github.com/stm32duino/BoardManagerFiles/raw/master/package_stmicroelectronics_index.json
- Navigate to: Tools > Board: "…" > Boards Manager…
- In the search box, type in "STM32". Afterwards a package with the name "STM32 MCU based boards (EtherCATduino)" should appear. Click install to finish adding the board.
- You can now select the EtherCATduino by going to: Tools > Board: "…" > "STM32 boards groups …", clicking "elecgator boards" and afterwards choosing "EtherCATduino" in Tools > Board Part Number



Preferences

Settings Network

Sketchbook location: C:\Users\piete\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation

Compiler warnings: None

□ Display line numbers

☑ Verify code after upload

☑ Check for updates on startup

□ Use accessibility features

Additional Boards Manager URLs: elecgator.com …e_stmicroelectronics_index.json

More preferences can be edited directly in the file

C:\Users\piete\AppData\Local\Arduino15\prefere…

(edit only when Arduino is not running)

Additional Boards Manager URLs

Enter additional URLs, one for each row

https://files.elecgator.com/EtherCATduino/package_stmicroelectr…

Boards Manager

Type All stm32

STM32 MCU based boards (EtherCATduino)

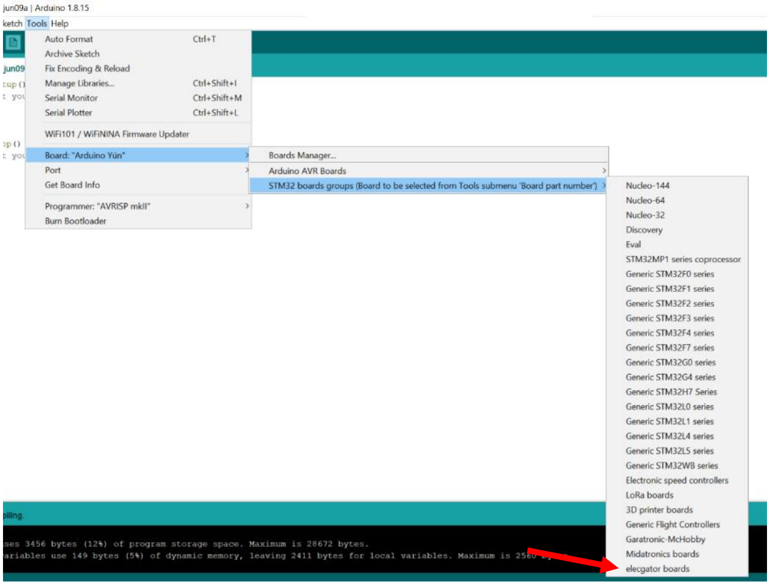
by **STMicroelectronics**

Boards included in this package:

Nucleo-144, Nucleo-64, Nucleo-32, Discovery, Eval, STM32MP1 series coprocessor, Generic STM32F0 series, Generic STM32F1 series, Generic STM32F2 series, Generic STM32F3 series, Generic STM32F4 series, Generic STM32F7 series, Generic STM32G0 series, Generic STM32G4 series, Generic STM32H7 Series, Generic STM32L0 series, Generic STM32L1 series, Generic STM32L4 series, Generic STM32L5 series, Generic STM32WB series, Electronic speed controllers, LoRa boards, 3D printer boards, Generic Flight Controllers, Garatronic-McHobby, Midatronics boards, elecgator boards.

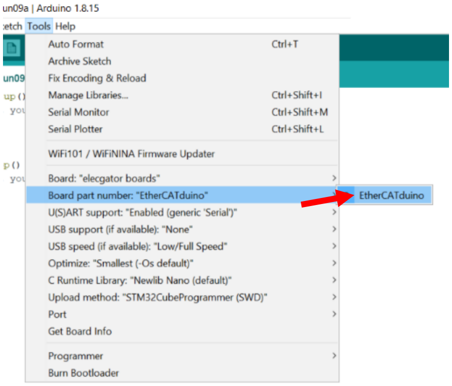
Online Help

More Info



jun09a | Arduino 1.8.15

```
ses 3456 bytes (12%) of program storage space. Maximum is 28672 bytes.
ariables use 149 bytes (5%) of dynamic memory, leaving 2411 bytes for local variables. Maximum is 2560
```



un09a | Arduino 1.8.15

etch Tools Help
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: "elecgator boards"
Board part number: "EtherCATduino"
U(S)ART support: "Enabled (generic 'Serial')"
USB support (if available): "None"
USB speed (if available): "Low/Full Speed"
Optimize: "Smallest (-Os default)"
C Runtime Library: "Newlib Nano (default)"
Upload method: "STM32CubeProgrammer (SWD)"
Port
Get Board Info
Programmer
Burn Bootloader
EtherCATduino

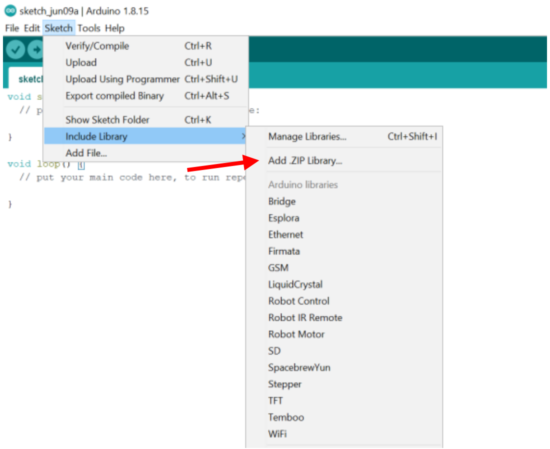
1.4.2 Programming

Before you can upload a sketch to the EtherCATduino, you first have to install the STM32CubeProgrammer utility. This utility can be download here.
To program an EtherCATduino with your Arduino sketch, you have to put the board in DFU mode. To do this, press and hold the reset button and afterwards press and hold the DFU button. Without releasing the DFU button, release the reset button. Now release the DFU button. The board will now be in DFU mode. The EC RUN and EC ERR LEDs will now light up slightly. This process has to be repeated before every upload. In the Arduino IDE, make sure to select "STM32CubeProgrammer (DFU)" as the upload method. You can now press the Upload button to upload your sketch to the EtherCATduino.

2 EtherCATduino Library

2.1 Installation in Arduino IDE

You can download the EtherCATduino library here. After downloading the library, you can add it to the Arduino IDE by navigating to Sketch > Include Library > Add .ZIP Library… and choosing the downloaded .ZIP file from your downloads folder.



2.2 Method Summary

Method	Parameters	Return type	Description
EtherCATduino	void	void	Constructor
~EtherCATduino	void	void	Destructor
begin	void	void	Place this function in the void setup() method of your Arduino sketch. This function initializes all peripherals to initiate data communication over EtherCAT. After this function has been run, everything is ready to further use the library
update	void	void	Place this function in the void loop() method of your Arduino sketch. This function polls for the EtherCAT outputs and updates the EtherCAT inputs. Make sure to run this function before retrieving data from the outputs and after updating the input data.
getInput	uint16_t index, uint8_t &value	int	Retrieve the input value for the specified index. The value will be saved in the parameter "value". Returns 1 when the index is invalid, otherwise returns 0.
setInput	uint16_t index, uint8_t value	int	Sets the input variable for the specified index. Returns 1 when the index is invalid, otherwise return 0.
getOutput	uint16_t index, uint8_t &value	int	Retrieve the output variable for the specified index. The value will be saved in the parameter "value".

			Returns 1 when the index is invalid, otherwise returns 0.
getMaxInputIndex	void	uint16_t	Returns the amounts of existing inputs. Useful when manipulating the inputs variable directly
getMaxOutputIndex	void	uint16_t	Returns the amounts of existing outputs. Useful when manipulating the outputs variable directly
getConfig1…8	void	uint32_t	Returns the config variable with the corresponding index.
setConfig1…8	uint32_t value	void	Sets the config variable with the corresponding index to parameter “value”.
getInfo1…8	void	uint32_t	Returns the info variable with the corresponding index.
setInfo1…8	uint32_t value	void	Sets the info variable with the corresponding index to parameter “value”.