



DEXTER BASE DOCUMENT

The purpose of this document is to familiarize yourself with the functionalities of DEXTER, a custom-built embedded processor built exclusively for your Build Club experiments

This document has three parts:

- **Dexter Quick Reference Guide**
- **STM IDE Installation guide**
- **Dexter User Functions**

(click the part you'd like to read)





DEXTER

an Embedded Controller Platform

Quick reference guide

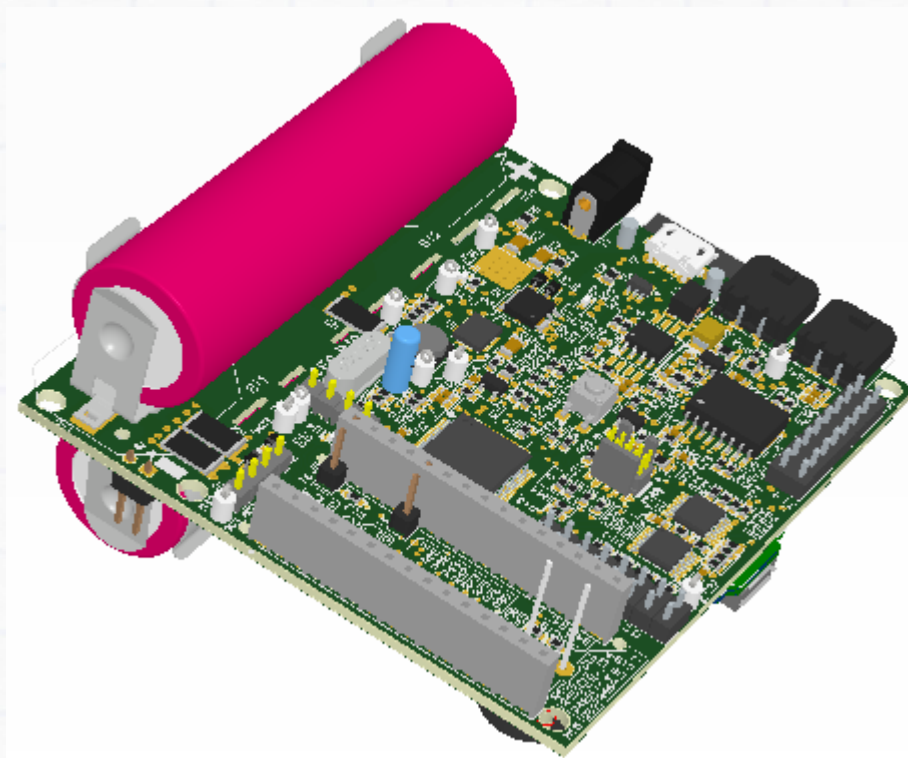




Table of Contents

- 1. DEXTER: Product View4
- 2. DEXTER : Hardware Architecture and Specification8
- 3. DEXTER : Memory, Indicators and Ports10
 - a) Memory10
 - b) LED status indicators10
 - c) Audio (Buzzer) indicator11
 - d) Ports.....11
- 4. DEXTER Power ON Procedure16
- 5. COTS Plugin options on Expansion slots17
- 6. IO mapping of STM32L452RCT6 controller18
- 7. DEXTER ADAPTER BOARD19
- 8. COTS plug-in Options on Expansion slots using Adapter/Carrier Board20
 - a) Illustration 1 COTS Nano Ethernet shield plugged onto Adapter board that plugged to DEXTER.....20
 - b) Illustration 2 COTS Nano Ethernet shield and UNO motor shield plugged onto Adapter board that plugged to DEXTER21

DEXTER: Product View

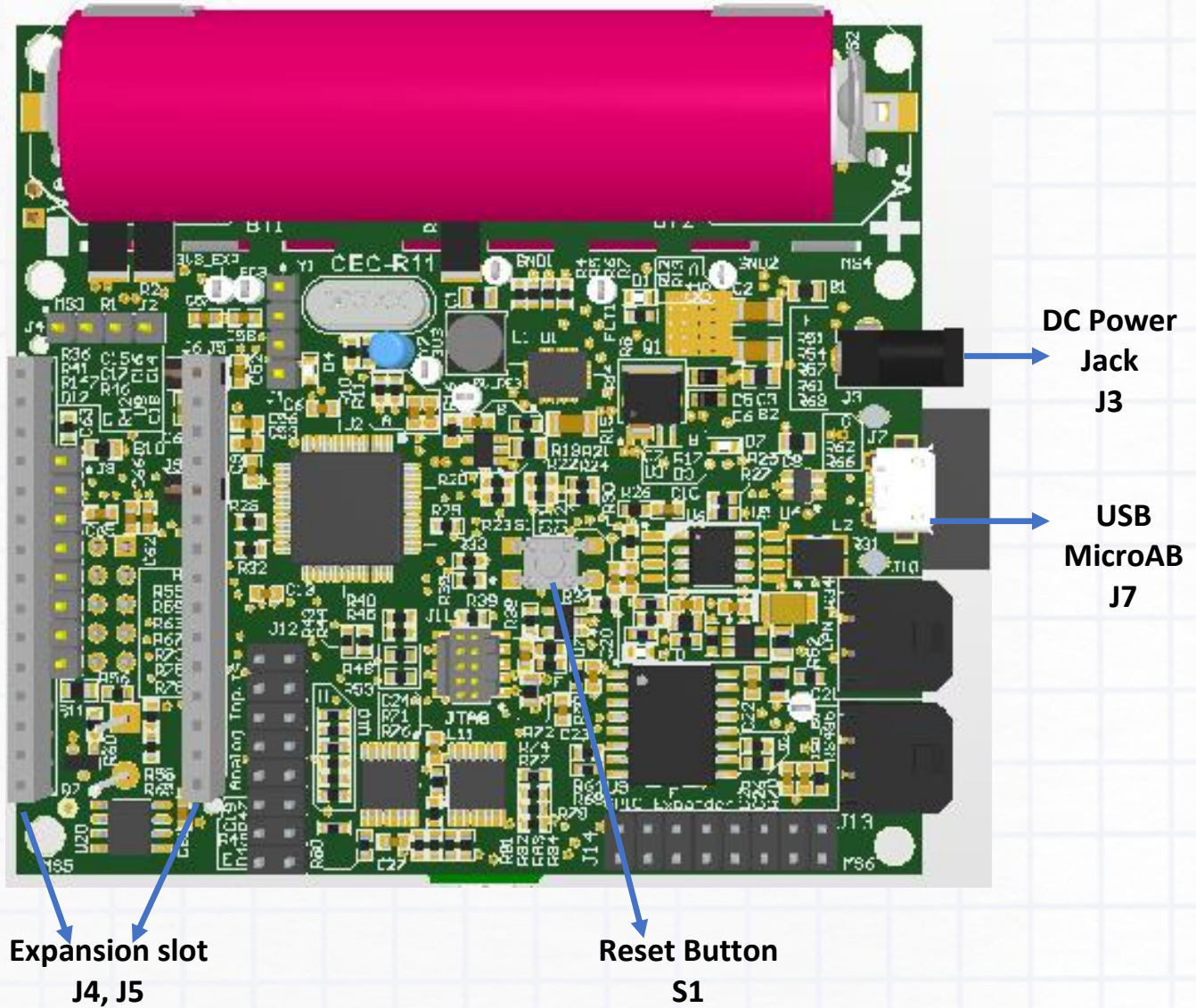


Figure 1: Top View



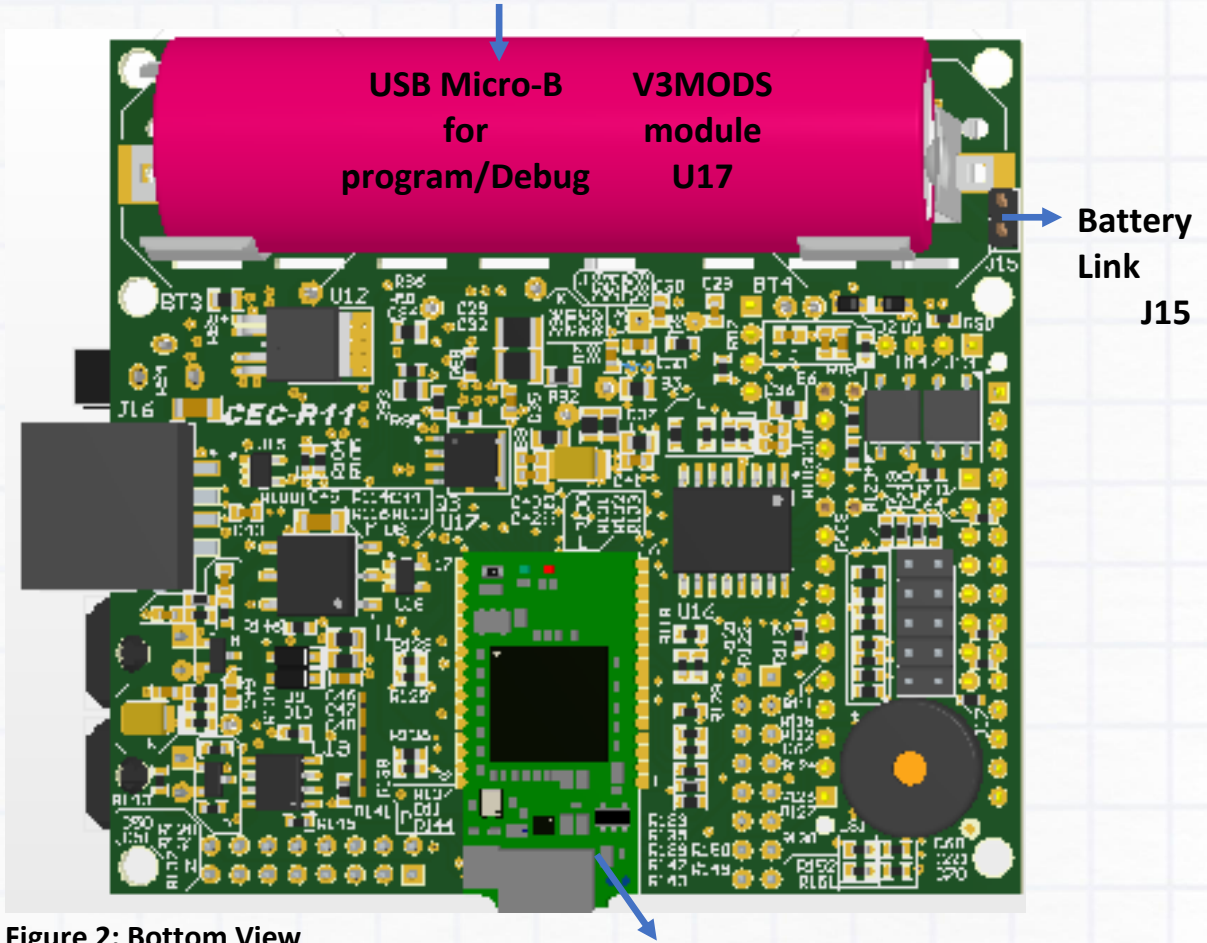


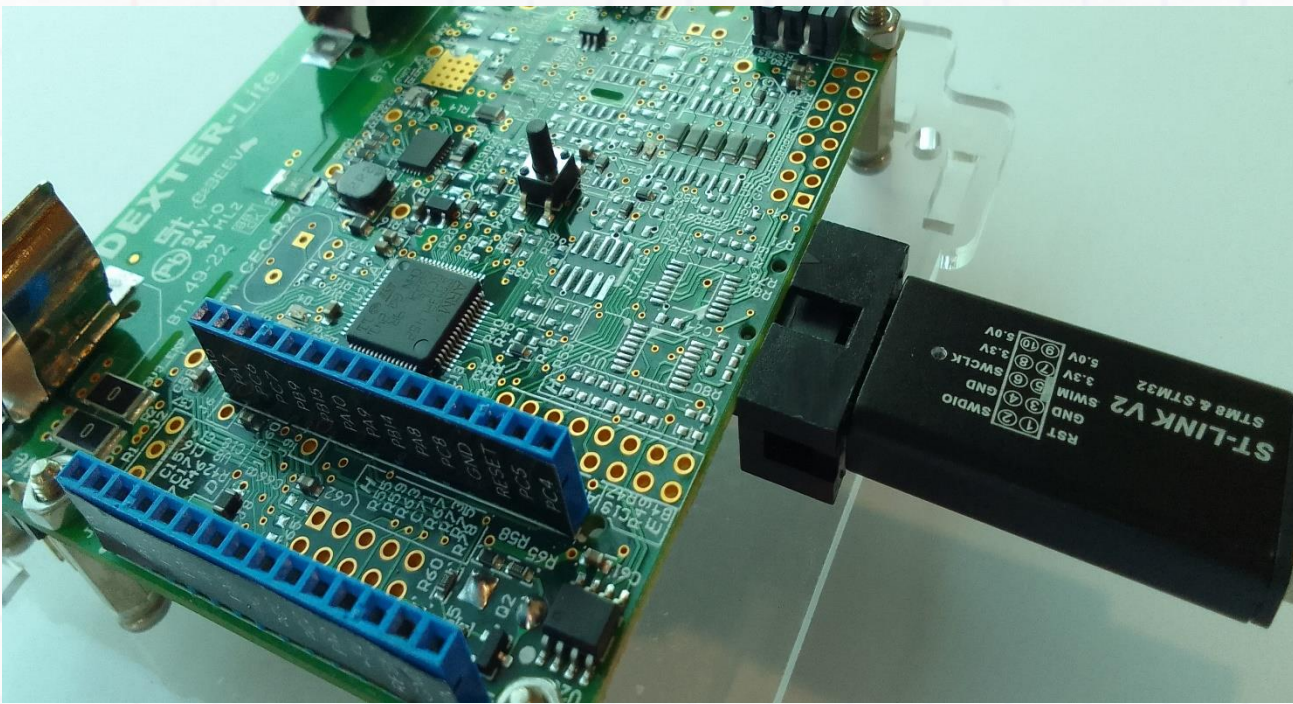
Figure 2: Bottom View

DEXTER -Lite

ST-LINK V2



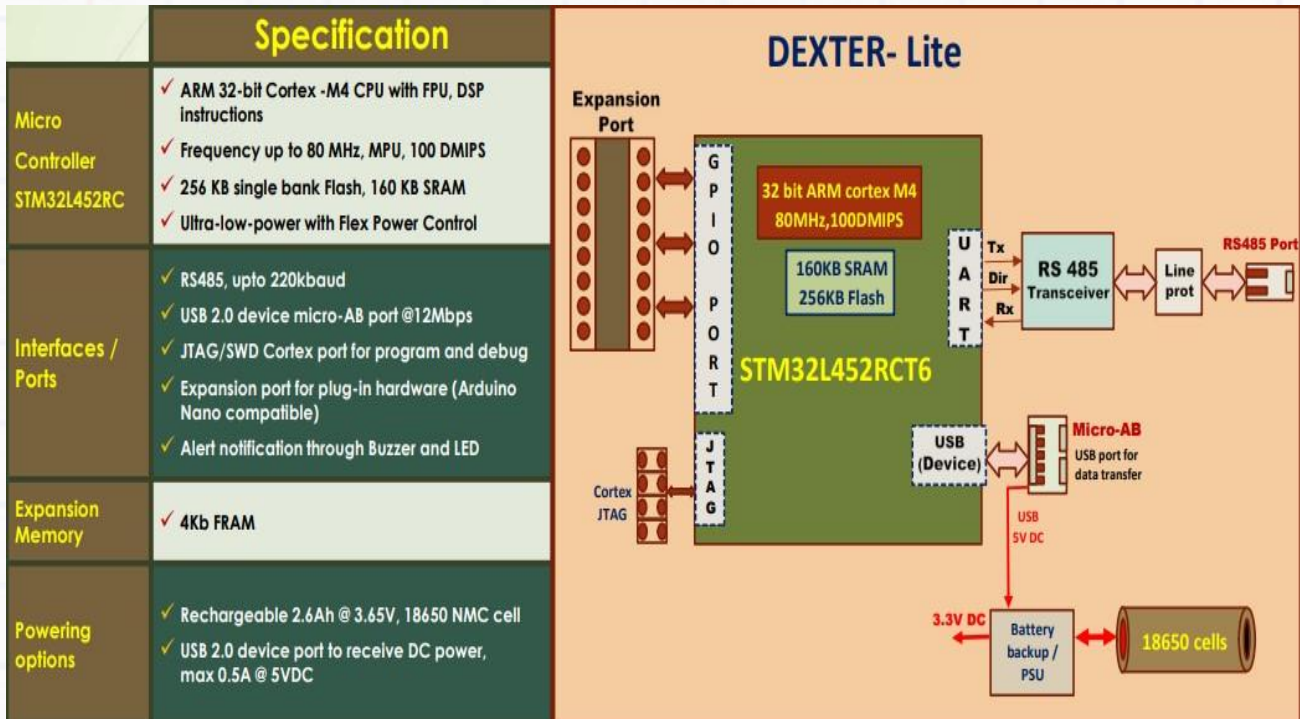
Connect this **ST-LINK V2** for flashing program to Dexter Lite Board





Please verify the ST-LINK has the same order as depicted in the figure.

DEXTER Lite: Hardware Architecture and Specification



DEXTER : Hardware Architecture and Specification

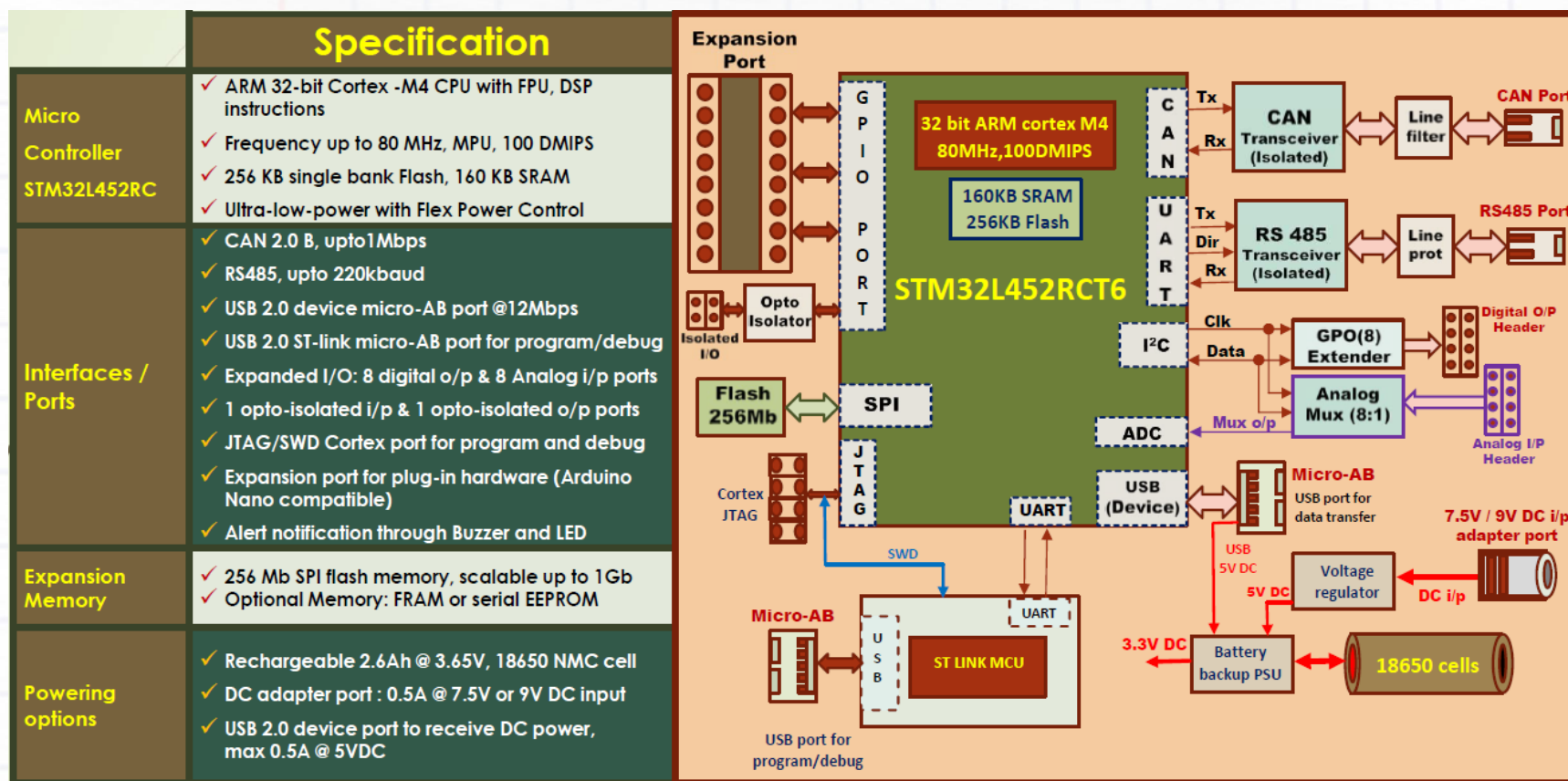
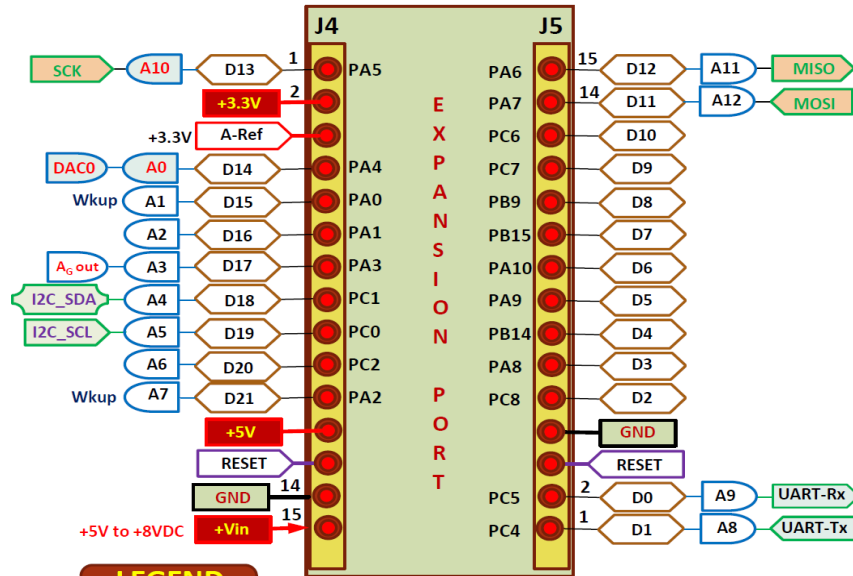


Figure 3: Block Diagram

Expansion port



LEGEND

Dx — Digital I/O (5V tolerant)	I2C_SDA — I2C Data I/O	UART-Rx — UART Data I/P to controller
Ax — Analog I/P to controller (3.3V tolerant)	I2C_SCL — I2C clock O/P from controller	UART-Tx — UART Data O/P from controller
DACx — Analog O/P from controller (3.3V tolerant)	MOSI — SPI Data O/P from controller	GND — DC negative
Ax — Analog I/P to controller (5V tolerant)	MISO — SPI Data I/P to controller	A-Ref — Analog Reference to peripheral cards
A ₀ out — OP-Amp O/P from controller	SCK — SPI clock O/P from controller	+V — DC positive
		RESET — Reset O/P to peripheral cards

Multifunctional Pin extensions

Digital I/O: 21 Ports

- ✓ Bidirectional ports
- ✓ 5V tolerant inputs
- ✓ I/O current @ 8mA
- ✓ 10 Timer channels
- ✓ 2 x Wakeup pins

Analog I/O: 12 Ports

- ✓ 12 x ADC input
- ✓ 1 x DAC output
- ✓ 1 x Comparator
- ✓ 1x OP-Amp with non-inv gain up to 16

Comm I/O: 3 serial interfaces

- ✓ 1 x I2C port @ 100 kbps to 1 Mbps data rate
- ✓ 1 x SPI port @ 40 Mbps max data rate
- ✓ 1 x UART port @ Up to 204 kbaud

- ✓ Plug-in support for Arduino Nano I/O cards
- ✓ Support Arduino Uno/MKR cards through carrier card

Power pins:

- ✓ + Vin: +5V to +8V DC, Up to 300mA @5V with a 18Wh battery backup
- ✓ Analog Ref: +3.3V
- ✓ I/O supply: +3.3V@100mA

DEXTER : Memory, Indicators and Ports

a) Memory

U16 : Flash Memory – SPI Interface

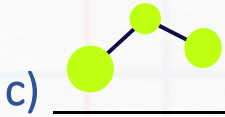
Pin #	Controller Pin mapping	Flash Memory Pin Name	Signal Name	Function
3	NRST	$\overline{\text{RESET}}$	FM_RST	Memory Reset
7	PA15	$\overline{\text{CS}}$	FM_CS	Chip Select
8	PC11	DO(IO1)	FM_MISO	Data I/P to Controller
9	PB5	$\overline{\text{WP}}(\text{IO2})$	FM_WP	Write Protect
15	PC12	DI(IO0)	FM_MOSI	Data O/P from Controller
16	PC10	CLK	FM_SCK	Clock Input

U20 : FRAM Memory – I2C1 Interface

Pin #	Controller Pin mapping	FRAM Memory Pin Name	Signal Name	Function
5	PB7	SDA	uC_I2C1_SDA	Serial Data I/O
6	PB6	SCL	uC_I2C1_SCL	Serial Clock

b) LED status indicators

LED Reference	Controller Pin mapping	Function
D1	-	Glows to indicate Battery Charging
D4	PH0	Blinks to indicate Controller activity
D7	-	Glows to indicate USB Power switch Fault
D12	-	Glows to indicate 5V Power



c)

c) Audio (Buzzer) indicator

Buzzer Reference	Controller Pin mapping	Function
LS1	PB3	Piezo electric buzzer (4KHz) with audible cadence

d) Ports

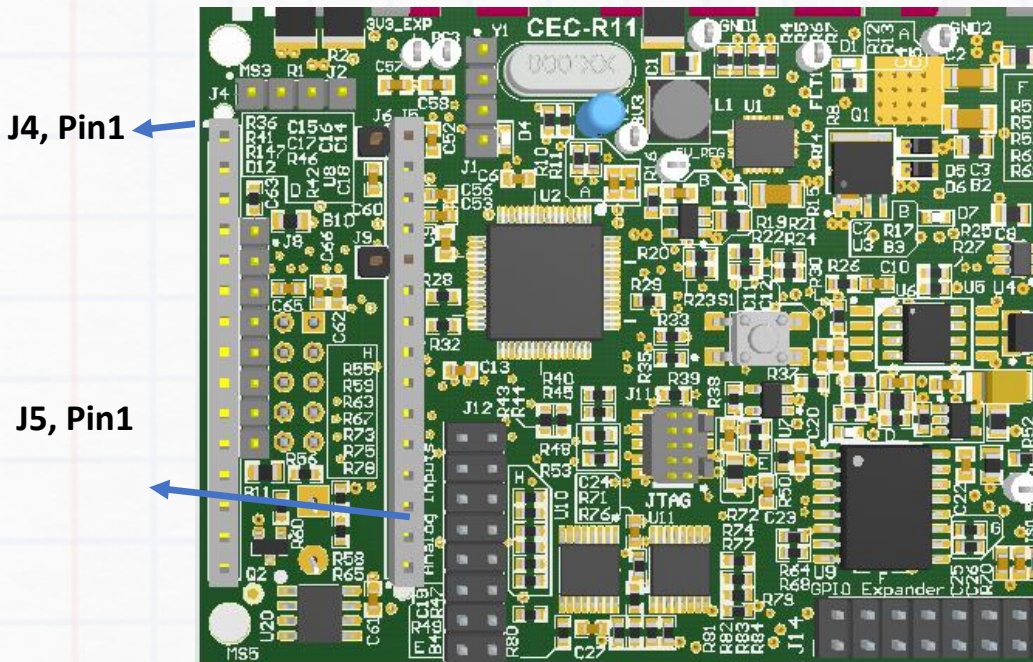


Figure 5: Expansion Port

J4 : Expansion Port connector Pinout

Table below summarizes the Pinout functions of Expansion Port shown in **Figure 4**.

Pin #	Controller Pin mapping	Connector Pin Name	Controller Pin Multi-function features
1	PA5	D13/A10/SCK	D13: Digital I/O
			A10: Analog I/P(3.3V tolerant)
			SCK: SPI Clock O/P from Dexter; To interface Dexter using SPI interface
2	-	+3.3V	+3.3V DC Out from Dexter
3	-	A-Ref	To connect to Analog Reference of ADD-ON cards, +3.3V
4	PA4	D14/A0/DAC0	D14: Digital I/O
			A0: Analog I/P (3.3V tolerant)
			DAC0: DAC O/P
5	PA0	D15/A1/Wkup	D15: Digital I/O
			A1: Analog I/P (5V tolerant)
			Wkup: Wakeup signal I/P
6	PA1	D16/A2	D16: Digital I/O
			A2: Analog I/P (5V tolerant)
7	PA3	D17/A3/A _G out	D17: Digital I/O
			A3: Analog I/P (5V tolerant)
			A _G out: Op-Amp O/P from Dexter
8	PC1	D18/A4/ I2C_SDA	D18: Digital I/O
			A4: Analog I/P (5V tolerant)
			I2C_SDA: I2C Serial Data I/O
9	PC0	D19/A5/ I2C_SCL	D19: Digital I/O
			A5: Analog I/P (5V tolerant)
			I2C_SCL: I2C Serial Clock O/P from Dexter



Pin #	Controller Pin mapping	Connector Pin Name	Controller Pin Multi-function features
10	PC2	D20/A6	D20: Digital I/O
			A6: Analog I/P (5V tolerant)
11	PA2	D21/A7/Wkup	D21: Digital I/O
			A7: Analog I/P (5V tolerant)
			Wkup:Wakeup signal I/P
12	-	+5V	+5V O/P from Adapter board
13	NRST	RESET	RESET
14	-	GND	Ground
15	-	+VIN	Power I/P (+5V to +8V) to Adapter board



J5 : Expansion slot connector Pinout

Pin #	Controller Pin mapping	Pin Name	Controller Pin Multi-function features
1	PC4	D1/A8/UART-Tx	D1: Digital I/O
			A8: Analog I/P (5V tolerant)
			UART-Tx: UART Data O/P from Dexter
2	PC5	D0/A9/UART-Rx	D0: Digital I/O
			A9: Analog I/P (5V tolerant)
			UART-Rx: UART Data I/P to Dexter
3	NRST	RESET	RESET O/P to ADD-ON cards
4	-	GND	Ground
5	PC8	D2	D2: Digital I/O
6	PA8	D3	D3: Digital I/O
7	PB14	D4	D4: Digital I/O
8	PA9	D5	D5: Digital I/O
9	PA10	D6	D6: Digital I/O
10	PB15	D7	D7: Digital I/O
11	PB9	D8	D8: Digital I/O
12	PC7	D9	D9: Digital I/O
13	PC6	D10	D10: Digital I/O
14	PA7	D11/A12/MOSI	D11: Digital I/O
			A12: Analog I/P (5V tolerant)
			MOSI: SPI Data O/P from Dexter
15	PA6	D12/A11/MISO	D12: Digital I/O
			A11: Analog I/P (5V tolerant)
			MISO: SPI Data I/P to Dexter



J7 : USB Type AB connector Pinout

Pin #	Controller Pin mapping	Pin Name	Function
1	-	VBUS	VBUS (+5V)
2	-	USB_N	USB Differential Pair M
3	-	USB_P	USB Differential Pair P
4	PA10	OTG_FS_ID	USB ID
5	-	GND	Ground

J11: SWD / JTAG connector Pinout

Pin #	Controller Pin mapping	Pin Name	Function
1	-	3V3_VDD	+3.3V
2	PA13	SWDIO /TMS	SWD I/O ; Test Mode select
3	-	GND	Ground
4	PA14	SWCLK /JTCK	SW clock ; JTAG clock signal to target CPU
5	-	GND	Ground
6	PB3	TDO	JTAG data O/P from target CPU
7	-	NC	-
8	PA15	TDI	JTAG data I/P of target CPU
9	PB4	TRST	NC
10	NRST	SWD_RST	Reset

J15 : Shorting Link

Shorting Link to be inserted in J15 when using 2 cells in parallel

<u>Cell Orientation</u>		
Mode	Orientation	J15
2 Series	BT3 -ve, BT4 +ve	OPEN
2 Parallel	BT3 +ve, BT4 -ve	SHORT

DEXTER Power ON Procedure

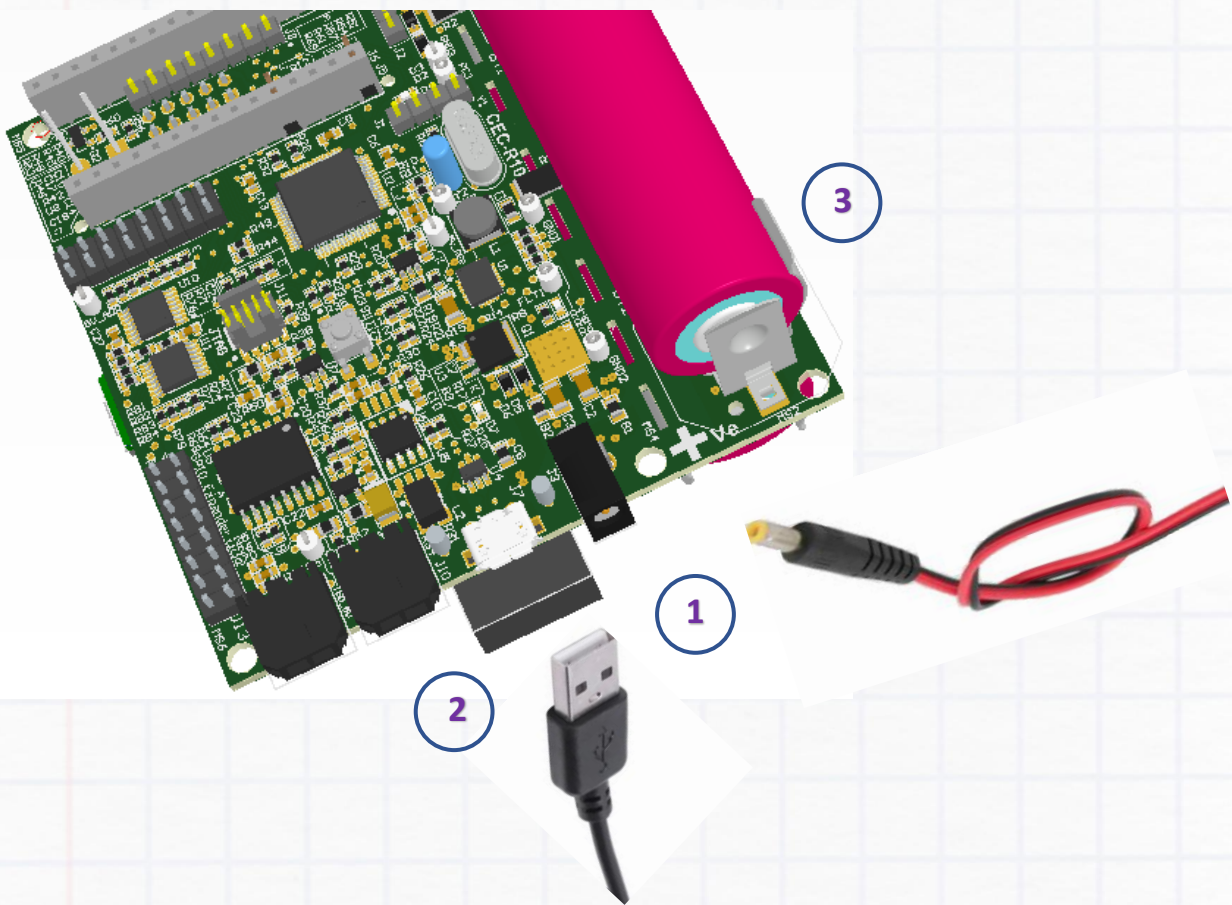


Figure 6: Powering options

DC Powering options

1. External 500mA @ 7.5V or 9V DC input from DC power adapter
2. USB 2.0 upto 500mA@5VDC
3. On-board 18650 rechargeable battery ; energy capacity of 9.5wh; on-board charger circuit

COTS Plugin options on Expansion slots

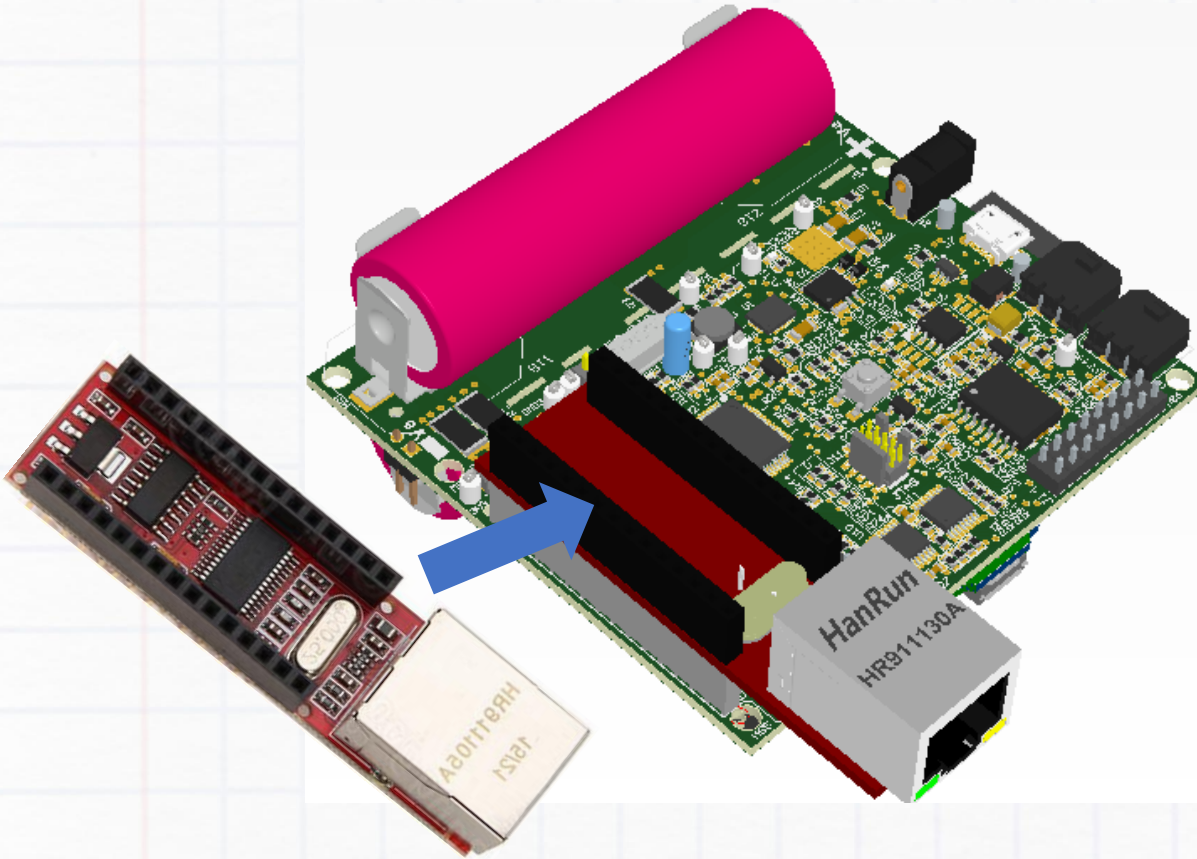


Figure 7: Plugging COTS boards

COTS (Commercially Off the shelf) Plug-in boards

Optional COTS boards on Expansion Slots	Applications
NANO IoT	IoT
NANO 33 BLE	Wi-Fi/BLE network connectivity.
NANO Ethernet Shield	Connect to Ethernet Network

IO mapping of STM32L452RCT6 controller

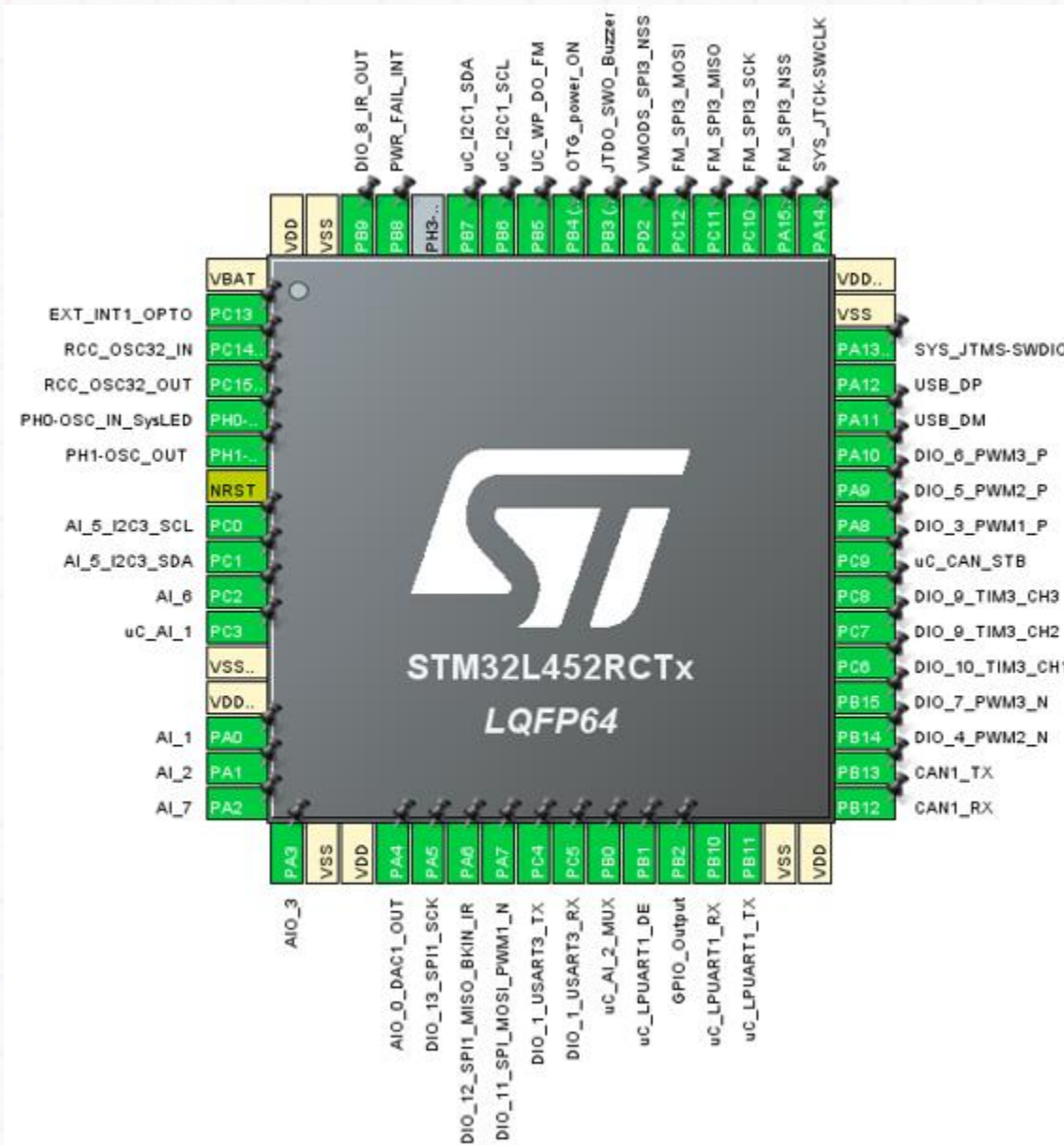


Figure 8: Pin mapping of DEXTER controller

DEXTER ADAPTER BOARD

Illustrations to expand Dexter efficacy using “Adapter Board” on expansion slots provided.

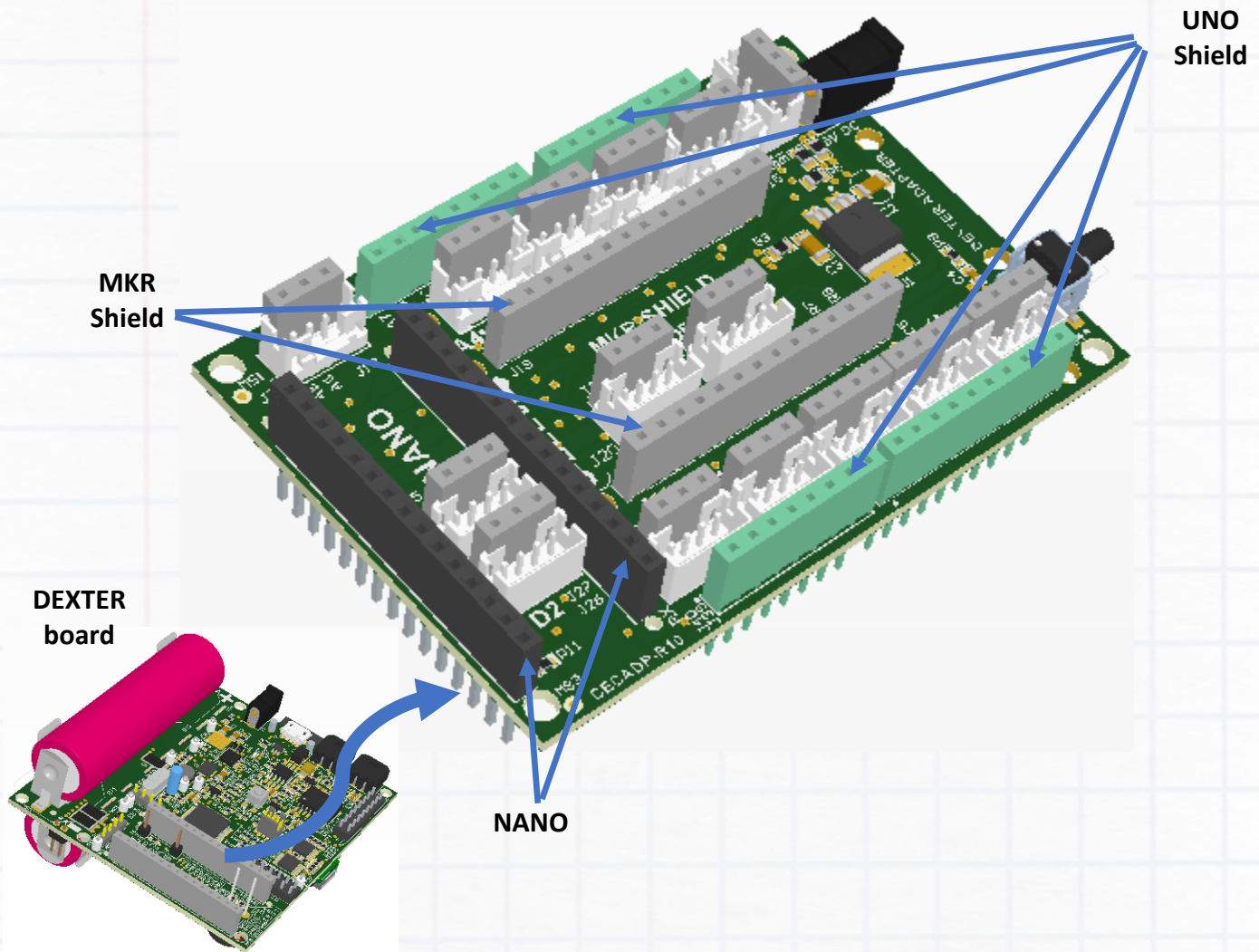


Figure 11: DEXTER Adapter board: attached on expansion slot of DEXTER

COTS plug-in Options on Expansion slots using Adapter/Carrier Board

a) Illustration 1 COTS Nano Ethernet shield plugged onto Adapter board that plugged to DEXTER

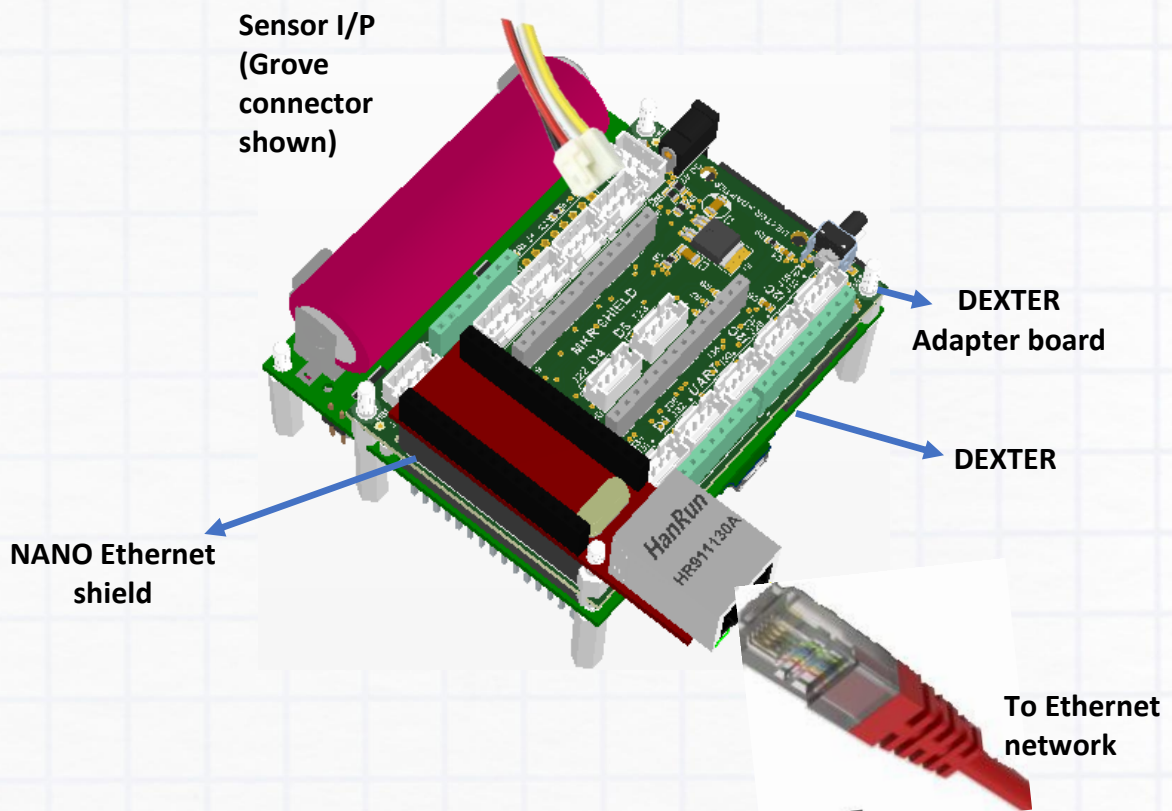


Figure 12: Illustration 1 *COTS Nano Ethernet shield plugged onto Adapter board that plugged to DEXTER*

b) Illustration 2 COTS Nano Ethernet shield and UNO motor shield plugged onto Adapter board that plugged to DEXTER

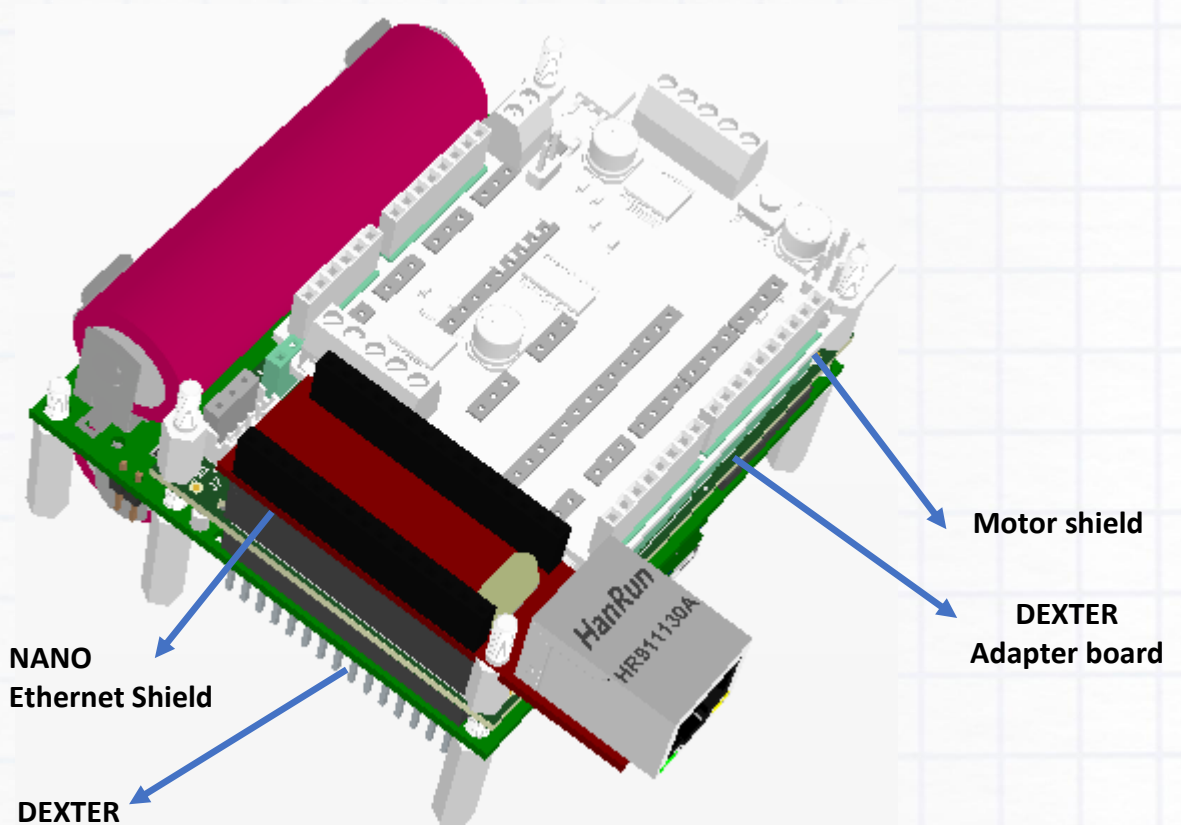


Figure 13: Illustration 2 *COTS UNO motor shield and Nano Ethernet shield plugged onto Adapter board that plugged to DEXTER*



STM IDE Installation guide



STM IDE Installation guide

- 1) Go to <https://www.st.com/en/development-tools/stm32cubeide.html#get-software>
Locate the installer for your operating system and click 'Get Latest'.

st.com/en/development-tools/stm32cubeide.html#get-software

All features

- Integration of services from STM32CubeMX: STM32 microcontroller, microprocessor, development platform and example project selectionPinout, clock, peripheral, and middleware configurationProject creation and generation of the initialization codeSoftware and middleware completed with enhanced STM32Cube Expansion Packages
- Based on Eclipse®/CDT™, with support for Eclipse® add-ons, GNU C/C++ for Arm® toolchain and GDB debugger

[Read more](#)

Get Software

Part Number	General Description	Supplier	Download	All versions
STM32CubeIDE-DEB	STM32CubeIDE Debian Linux Installer	ST	Get latest	Select version
STM32CubeIDE-Lnx	STM32CubeIDE Generic Linux Installer	ST	Get latest	Select version
STM32CubeIDE-Mac	STM32CubeIDE macOS Installer	ST	Get latest	Select version
STM32CubeIDE-RPM	STM32CubeIDE RPM Linux Installer	ST	Get latest	Select version
STM32CubeIDE-Win	STM32CubeIDE Windows Installer	ST	Get latest	Select version

Click 'Get Latest' for your operating system

Feedback



2) Now a pop up would appear to accept STM32's terms and agreements. Click 'Accept'

Get Software

If you have an account on my.st.com, login and download the software without any further validation steps.

[Login/Register](#)

If you don't want to login now, you can download the software by simply providing your name and e-mail address in the form below and validating it.

This allows us to stay in contact and inform you about updates of this software.

For subsequent downloads this step will not be required for most of our software.

First Name:

Last Name:

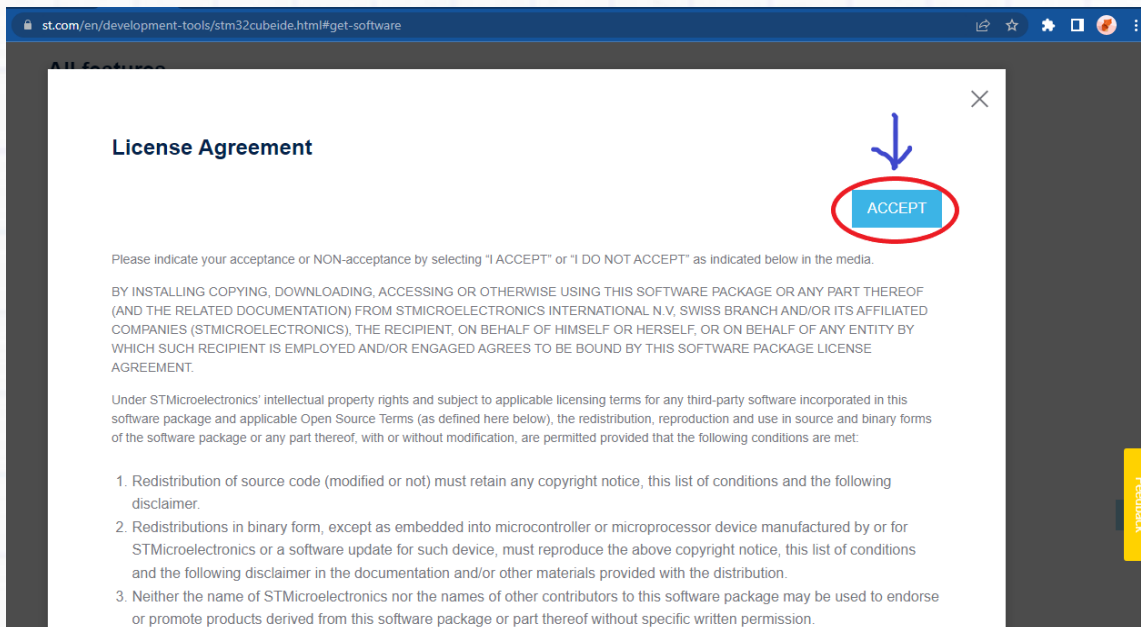
E-mail address:

Please review our [Privacy Statement](#) that describes how we process your profile information and how to assert your personal data protection rights

Please keep me informed about future updates for this software or new software in the same category

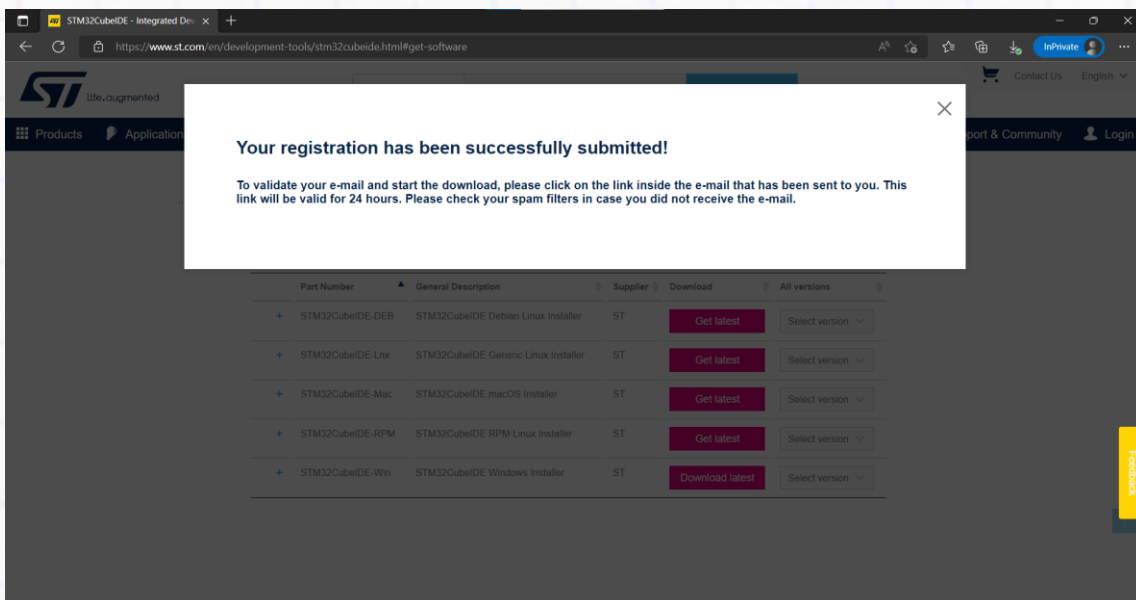
[Download](#)

3) Enter your details in the next window as shown. Next, hit 'Download'.





- 4) A confirmation pop up will be displayed conveying that an email with the Download link has been sent to the mail ID entered.

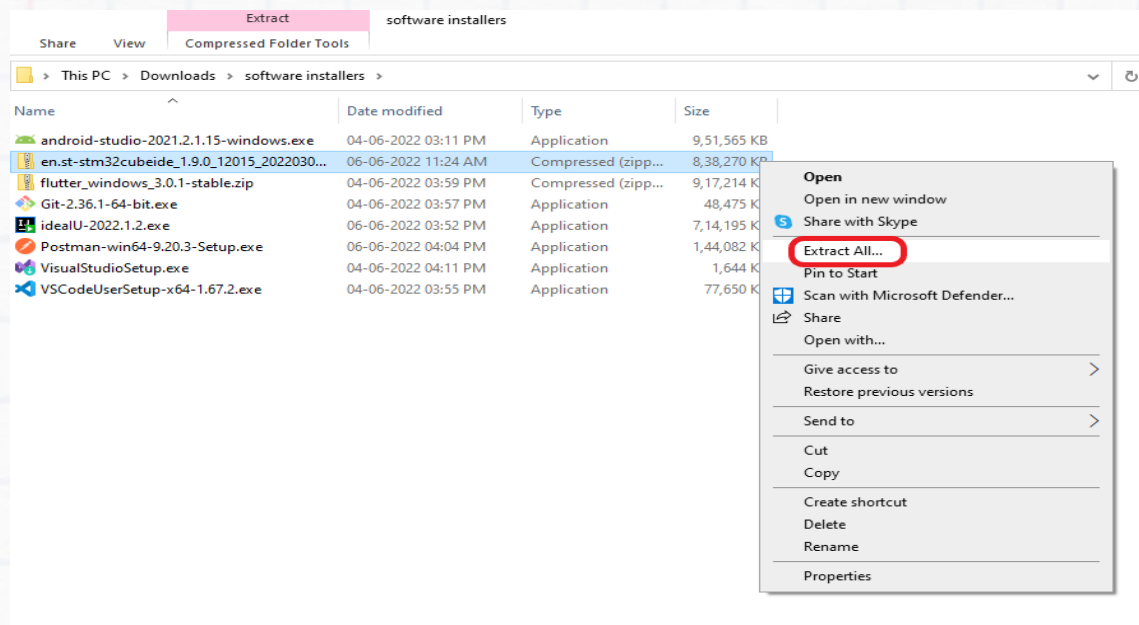




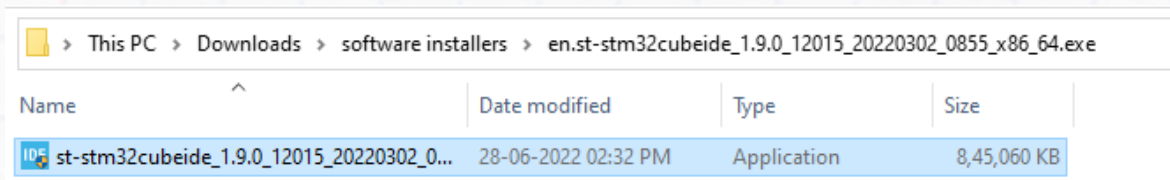
- 5) Go to your email and check for an email from STM32. Click on the **‘Download Now’** button in the email.



- 6) A zip file will start downloading. Once the download is completed, right-click the ZIP file and select 'Extract All'. The extracted folder will open when completed.



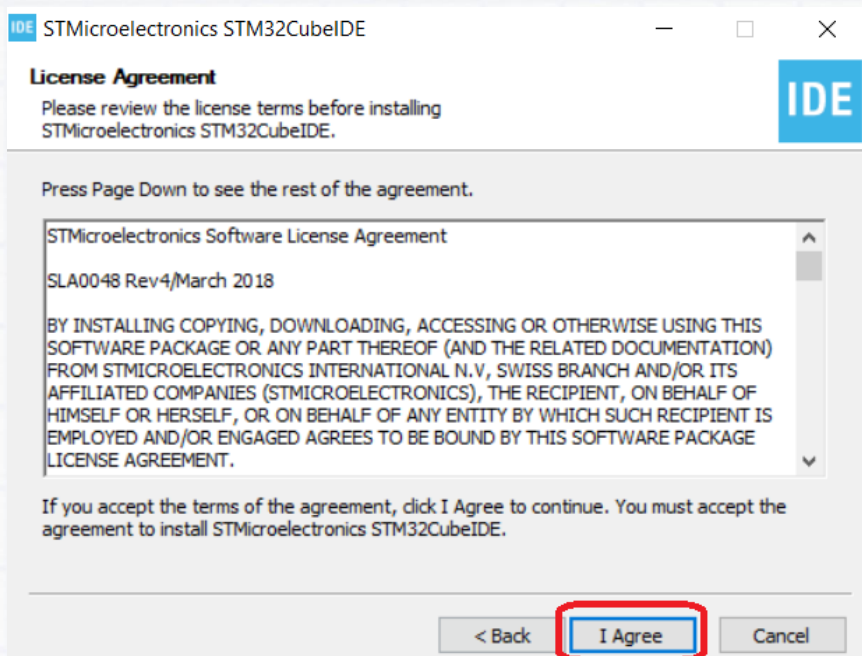
- 7) Inside the extracted folder, double-click the installer file.



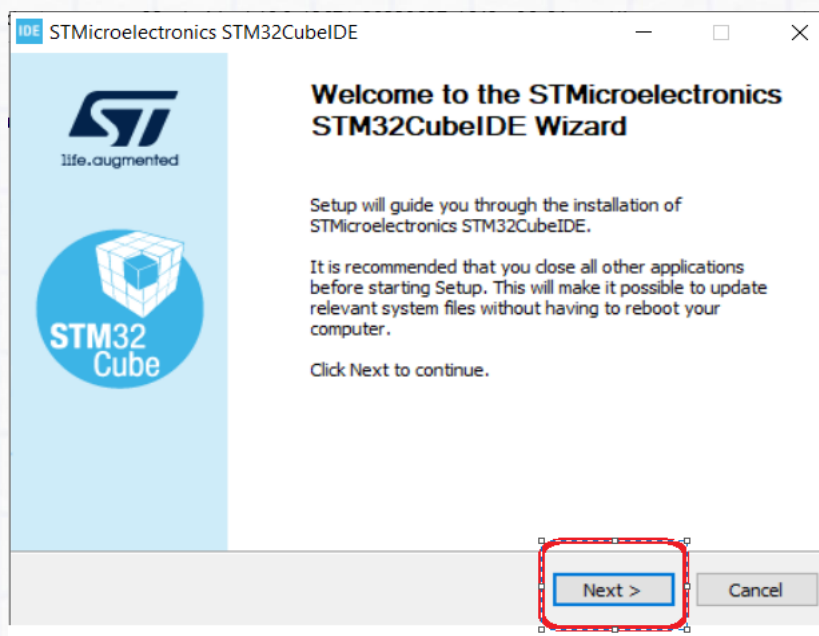
- 8) Follow the below steps:

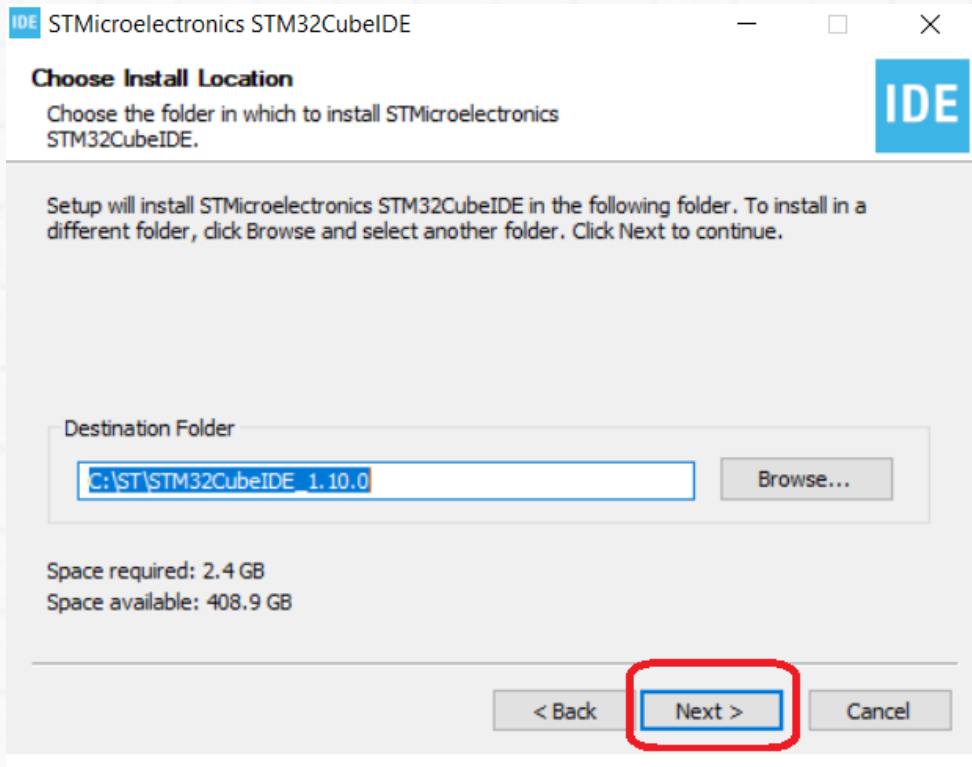


a)

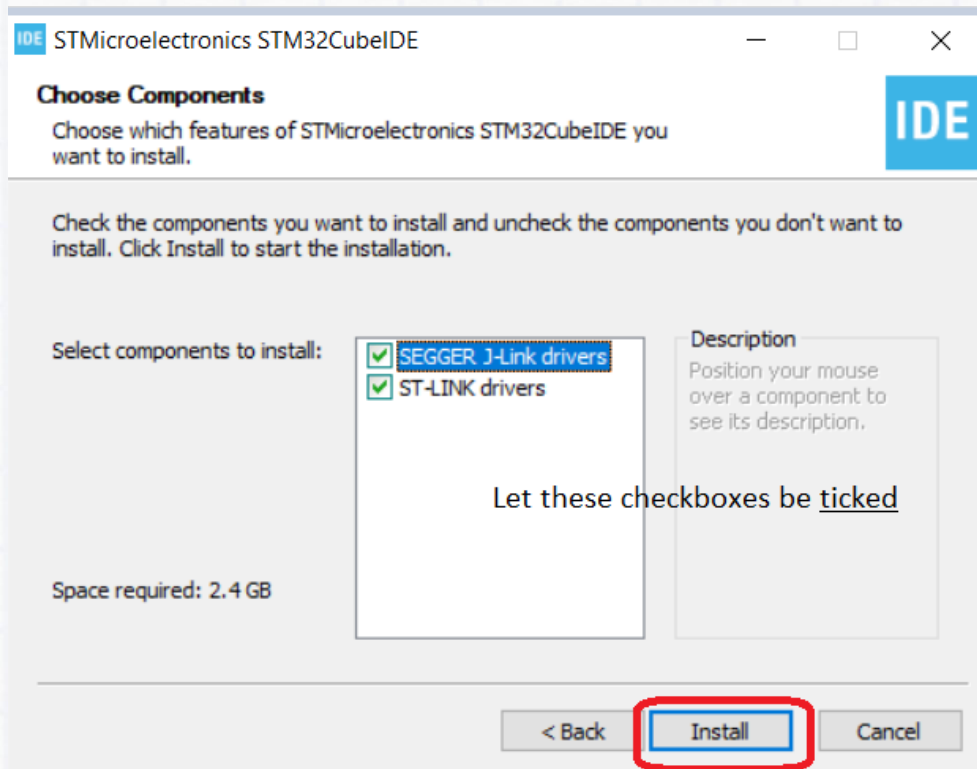


b)



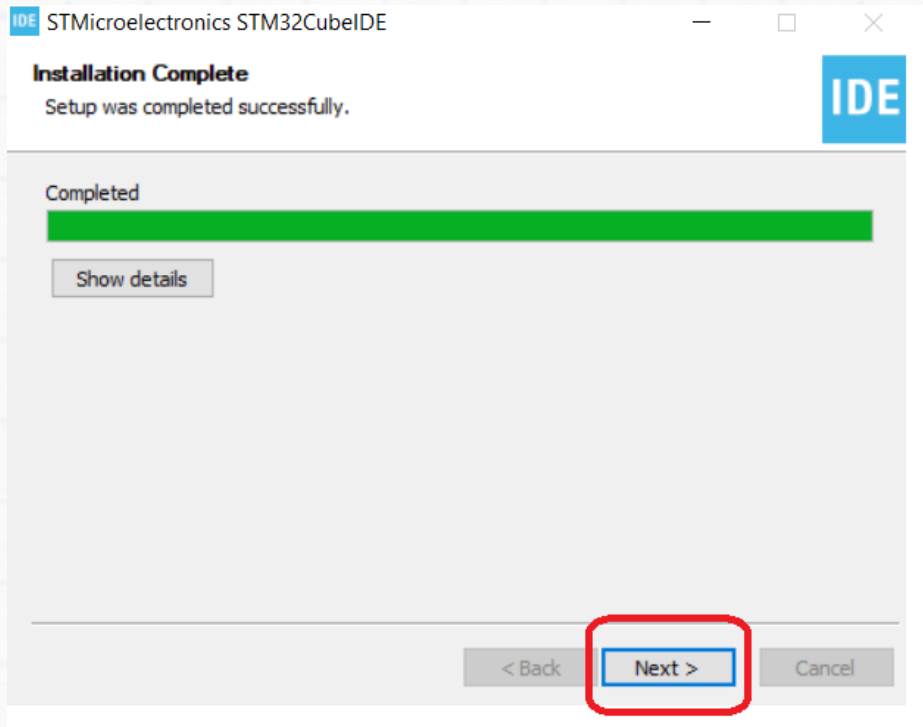


c)

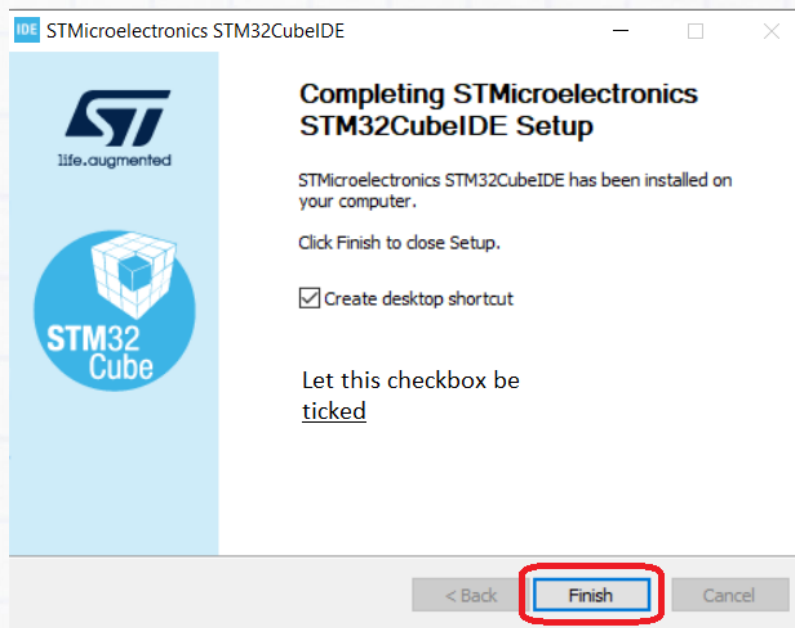




d)



e) Great! You have successfully downloaded & installed the STM32 Cube IDE software!

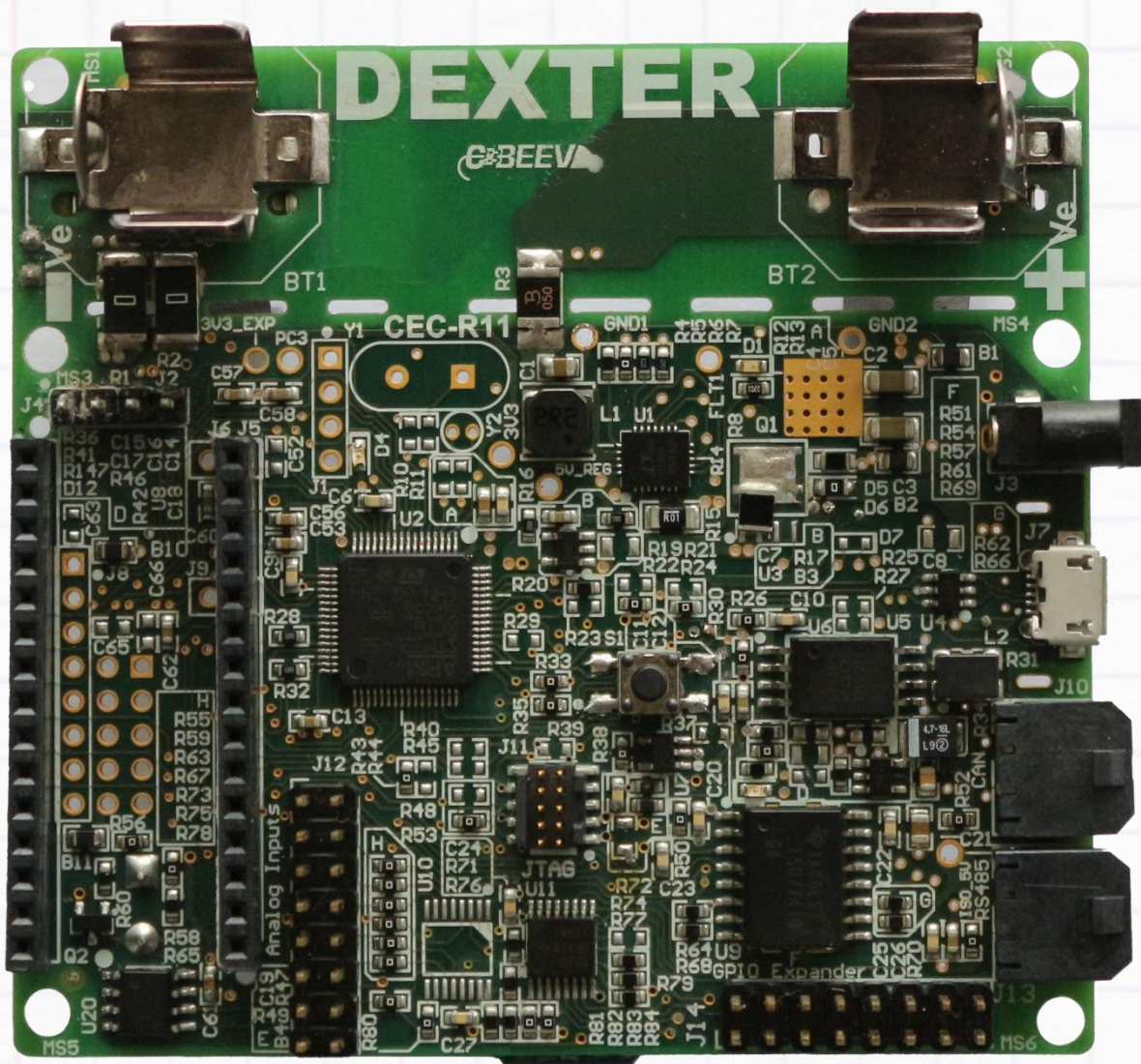




Dexter User Functions

DEXTER – User Functions

Base Document v1.0



The **Dexter Board** is designed using the microcontroller **STM32L452RCT6** (LQFP64 package) from STMicroelectronics. It is a versatile microcontroller, which can be used in a variety of ways. For the early Build Club experiments, designed for Engineering and Science students, irrespective of disciplines, Dexter ‘Wrapper’ functions have been predefined. These wrapper functions use the **STM32 CUBE IDE** (Integrated Development Environment). The IDE comes with certain default settings. Although it is possible to change these default settings, for the initial Build Club experiments we recommend that only default settings be used.

The Wrapper functions are as follows:

1) GPIO - General Purpose Input / Output

To program GPIO pins. Can be used either for Digital signals or Analog signals. They can be thus used for driving LEDs, Motors or receive signals from sensors.

Type	Function call	Description
Digital IO	Digital_transmit (Pin, State);	To turn ON/OFF Digital Output pins
	Digital_transmit_pwm (Pin, Duty cycle);	To transmit to an output using PWM
	Digital_receive (Pin);	To receive Digital Input pins
Analog IO	Analog_transmit (Pin, State);	To turn ON/OFF Analog Output pins
	Analog_receive (Pin);	To receive Analog Input pins

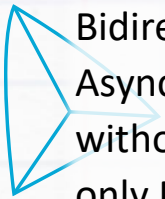
These are general functions that allow for managing the program’s flow control and inserting time delays in it.

Type	Function call	Description
Delay Timer	Delay_ms (Time);	To insert a time delay (in milliseconds) into the program flow
Setup	Dexter_setup()	For writing one-time setup/initialisation code
Loop	Dexter_loop()	Any code written inside this function will be executed in an infinite loop



Communication Interfaces

UART - Universal Asynchronous Receiver / Transmitter



Bidirectional serial ports used in Build Club experiments in an Asynchronous (data bytes transmitted between start and stop bits without a clock signal) as well as Half Duplex (can either only Transmit or only Receive at a given time) mode.

The Baud Rate of the channel is configurable.

Channel	Function call	Description
UART1	Uart1_init (baud rate)	To initialize UART1 with Baud Rate
	Uart1_receive ()	To receive data via UART1 using PB6 (TX), PB7 (RX)
	Uart1_receive_IT ()	To receive data via UART1 using PB6 (TX), PB7 (RX) through Interrupt mechanism
	Uart1_transmit ()	To transmit data via UART1 using PB6 (TX), PB7 (RX)
UART2	Uart2_init (baud rate)	To initialize UART2 with Baud rate
	Uart2_receive ()	To receive data via UART2 using PA2 (TX), PA3 (RX)
	Uart2_receive_IT ()	To receive data via UART2 using PA2 (TX), PA3 (RX) through Interrupt mechanism
	Uart2_transmit()	To transmit data via UART2 using PA2 (TX), PA3 (RX)
UART3	Uart3_init (baud rate)	To initialize UART3 with Baud rate
	Uart3_receive()	To receive data via UART3 using PC4 (TX), PC5 (RX)
	Uart3_receive_IT ()	To receive data via UART3 using PC4 (TX), PC5 (RX) through Interrupt mechanism
	Uart3_transmit()	To transmit data via UART3 using PC4 (TX), PC5 (RX)

I2C - Inter Integrated Circuit

A Synchronous (data is sent at every tick of the serial clock SCL being transmitted by the controller) serial communication protocol that uses a Half Duplex link.

Channel	Function Call	Description
I2C1 (Master)	I2C_receive()	To receive data via I2C using PB8 (SCL), PB9 (SDA)
	I2C_transmit()	To transmit data via I2C using PB8 (SCL), PB9 (SDA)

SPI - Serial Peripheral Interface

A Synchronous (data is sent at every tick of the serial clock SCL being transmitted by the controller) serial communication protocol that uses a Half Duplex link.

Channel	Function call	Description
SPI1 (Full Duplex Master)	SPI1_transmit()	To transmit data via SPI1 using PA1 (SCK), PA11 (MISO), PA12 (MOSI).
	SPI1_receive()	To receive data via SPI1 using PA1 (SCK), PA11 (MISO), PA12 (MOSI).
	SPI1_transmit_receive()	To receive and transmit data via SPI1 using PA1 (SCK), PA11 (MISO), PA12 (MOSI).
SPI2 (Full Duplex Slave)	SPI2_transmit()	To transmit data via SPI2 using PB10 (SCK), PC2 (MISO), PC3 (MOSI).
	SPI2_receive()	To receive data via SPI2 using PB10 (SCK), PC2 (MISO), PC3 (MOSI).
	SPI2_transmit_receive()	To receive and transmit data via SPI2 using PB10 (SCK), PC2 (MISO), PC3 (MOSI).
SPI3 (Full Duplex Master)	SPI3_transmit()	To transmit data via SPI3 using PC10 (SCK), PC11 (MISO), PC12 (MOSI).
	SPI3_receive()	To receive data via SPI3 using PC10 (SCK), PC11 (MISO), PC12 (MOSI).
	SPI3_transmit_receive()	To receive and transmit data via SPI3 using PC10 (SCK), PC11 (MISO), PC12 (MOSI).



THE END.

Contact :

buildclub@respark.iitm.ac.in