

VISHAY SEMICONDUCTORS

www.vishay.com

Optical Sensors

Application Note

Using The Software Code Library

By Hakimi Wanyusof

OVERVIEW

This application note provides an overview of our C / C++ software code library, which is available for all of our optical digital sensors except for VCNL4010, VCNL4020, and VCNL3020. There are dedicated software code and application note for these optical sensors, which can be found <u>here</u>. This application note explains the contents, the structure, and how to use the software code library. The software code library can be found under the "Design Tools" tab for each of the product page of the digital sensors in the website. The proximity sensor VCNL4035X01 will be used as an example for this application note. The software code library for this proximity sensor can be found <u>here</u> under the "Design Tools" tab.

THE CONTENTS OF THE SOFTWARE CODE LIBRARY

The software code file contains "C", "C++", and "Project Examples" folders. The "C" and "C++" folders contain the raw software code, which can be exported into the IDE software application of your selected microcontroller.



Fig. 1 - The Contents of the Software Code

www.vishay.com

Vishay Semiconductors

The DNA of tech™

Using The Software Code Library

Image: Image	are View				
Pin to Quick Copy Paster access Clipboar	Cut Copy path Paste shortcut	Move to* Copy to* Orga	Delete Rename	New item • Easy access • New folder	Properties • Open • Edit History Open
← → ~ ↑] >	VCNL4035X01 >	C			
> 🛃 Quick access	Name	^		Date modified	Туре
A Quick access	PSoC Cypres	s		3/16/2021 2:03 PM	File folder
> 🔷 OneDrive	STM32F			3/16/2021 2:03 PM	File folder
> SharePoint	l2C_Function	S.C		2/1/2021 9:57 PM	C Source
Shareronne	道 I2C_Function	s.h		2/1/2021 10:00 PM	C/C++ Header
> 🤰 This PC	🛗 typedefinitio	n.h		2/1/2021 10:08 PM	C/C++ Header
> 🥩 Network	団 VCNL4035X ()1.h		2/1/2021 10:00 PM	C/C++ Header
	CNL4035X0	1_ALS.c		2/1/2021 9:57 PM	C Source
> 📙 VCNL4035X01	CNL4035X0)1_Application_Lib	orary.c	2/1/2021 10:26 PM	C Source
	li VCNL4035X0	1_Application_Lib	orary.h	2/1/2021 10:28 PM	C/C++ Header
	li VCNL4035X0)1_Prototypes.h		2/1/2021 10:00 PM	C/C++ Header
	CNL4035X0	01_PS.c		2/1/2021 9:57 PM	C Source

Fig. 2 - The Contents of the "C" Folder

Image: Image	are View			
Pin to Quick Copy Paster access	Cut Copy path Paste shortcut	Move Copy to * to *	New item ▼ 1 Easy access ▼ 1 New 1 Folder	Properties
Clipboar	ď	Organize	New	Open
← → ~ ↑] >	VCNL4035X01 > 0	2++		
Quick access	Name	\sim	Date modified	Туре
	📕 Main		3/16/2021 2:03 PM	File folder
OneDrive	[] I2C_Function	s.cpp	2/1/2021 11:12 PM	C++ Source
SharePoint	li I2C_Function	s.h	2/1/2021 10:00 PM	C/C++ Header
	🗄 typedefinitio	n.h	2/1/2021 11:09 PM	C/C++ Header
Itis PC	li VCNL4035X0)1.h	2/1/2021 10:00 PM	C/C++ Header
Network	U VCNL4035X0	01_ALS.cpp	2/1/2021 11:12 PM	C++ Source
- Hetholik	U VCNL4035X0	1_Application_Library.cpp	2/1/2021 11:12 PM	C++ Source
VCNL4035X01	li VCNL4035X0	1_Application_Library.h	2/1/2021 10:28 PM	C/C++ Header
	li VCNL4035X0	01_Prototypes.h	2/1/2021 10:00 PM	C/C++ Header
	[[™] VCNL4035X0	01_PS.cpp	2/1/2021 11:12 PM	C++ Source

APPLICATION NOT

Fig. 3 - The Contents of the "C++" Folder



The DNA of tech."

Using The Software Code Library

On the other hand, the **"Project Examples"** folder contains the project files, which can be directly run by clicking the project / workspace files. The project files are the fastest way to start testing with our sensors, where the software code could be directly executed, given that the programmer has set up the same pinouts as the one, which has been set up in the project files. Currently, only STM32F, PSoC[®] Cypress, and Arduino microcontroller families will be supported for this project examples.

📕 🛃 📒 🖛 Project	Examples				
File Home Sh	are View				
Pin to Quick Copy Access	Cut Copy path Paste shortcut	Move Copy to * to *	Delete Rename	► Rew item Easy access New folder	Properties
Clipboa	rd	Orga	nize	New	Open
← → • ↑ 🖡 >	VCNL4035X01 > P	roject Examples			
📌 Quick access	Name	^		Date modified	Туре
	📕 Arduino			3/16/2021 6:31 PM	File folder
OneDrive	PSoC Cypress			3/16/2021 6:17 PM	File folder
📜 SharePoint	STM32F			3/16/2021 2:03 PM	File folder

Fig. 4 - The contents of the "Project Examples" Folder

THE STRUCTURE OF THE SOFTWARE CODE LIBRARY



Fig. 5 - The Structure of the Software Code Library

The provided software code, which consists of MCU-specific software code and sensor API as shown in Fig. 5, is designed to – be modular. It means that the programmer can use the sensor API regardless of microcontroller platforms. Only MCU specific O software code has to be added / ported or just use the provided code for the supported platforms.

3



The DNA of tech."

Using The Software Code Library

This modular approach will reduce the time during the application development regardless of the platform used because only the MCU-specific software code has to be added / ported for the whole code to work. For example, the programmer can use/port the examples code in **Main.c/ino** (For PSoC[®] Cypress or Arduino platform), **Main_User.c** (For STM32F platform), and **"Part Name"_Application_Library.c/cpp** to quickly initialize different sensor modes, printing sensor register values via COM Port for debugging purpose as well as some useful application functions like calculating Lux for the ALS sensor and calculating CCT value using the empirical approach for the color sensor.

The following platforms are supported by the MCU-specific software code:

- PSoC[®] Cypress
- Arduino and Arduino-supported MCU
- STM32F

Notes

- Vishay does not own the MCU-specific code. The MCU-specific codes, which have been modified, are owned either from Cypress (now Infineon AG) and STMicroelectronics under the condition that the codes have to be used with their respective microcontrollers. The MCU-specific code from Arduino can be used, distributed and / or modified under the terms of the GNU Lesser General Public License as published by the Free Software Foundation
- For Arduino and Arduino-supported MCU The codes have been tested with Arduino Uno and Teensy3.5. Other Arduino and Arduinosupported MCU platforms could still work or some small code adjustment might be needed especially with regards to **Wire.h/i2c_t3.h** library compatibility

Any other unsupported platforms would require the addition of API library code in **I2C_Functions.h** and **I2C_Functions.c/cpp**.

The comparison of the provided software code files between MCUs and product lines are shown in Fig. 6 and Fig. 7 respectively:



Fig. 6 - The Comparison of the Provided Software Code Files Between MCUs 4

Revision: 05-Aug-2021

Р Р

υ

LICATION

Z O

For technical questions, contact: <u>sensorstechsupport@vishay.com</u> THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT <u>www.vishay.com/doc?91000</u>



Using The Software Code Library



Fig. 7 - The Comparison of the Provided Software Code Files Between MCUs



The DNA of tech."

Using The Software Code Library

HOW TO USE THE SOFTWARE CODE STEP BY STEP

1. STM32F

- Step 1: Create a new project in STM32 Cube IDE
- Step 2: Configure the pinouts for I2C, RCC, USB_Device, USB_OTG_FS
- Step 3: Configure the corresponding USB CDC clock configuration for debugging via COM port purpose
- Step 4: Open the "C" folder
- Step 5: Add the following files into the STM32 Cube IDE or other supported IDE:
 - Main.c (From the "STM32F" folder)

Note

- Main.c will be updated to a default code by the STM32 Cube IDE. The three lines of codes (INIT_"Part Name"(), INIT_"Part Name"(), and PRINT_"Part Name"(), which call the required header file as well as the initialization and printing functions in Main_User.c, have to be written again when Main.c is updated to the default code after relaunch. The provided Main.c file can be used as an example of where to add this code if a code rewrite happens
- Main_User.c (From the "STM32F" folder)
- I2C_Functions.c
- "Part Name"_Application_Library.c
- "Part Name"_PS.c
- "Part Name"_ALS.c
- "Part Name"_RGB.c
- typedefinition.h
- Main_User.h (From the "STM32F" folder)
- I2C_Functions.h
- "Part Name"_Application_Library.h
- "Part Name".h
- "Part Name"_Prototypes.h
- Step 6: Activate the MCU-specific code in the files I2C_Functions.c, I2C_Functions.h, and "Part Name"_Application_ Library.c by defining #define STM32F in typedefinition.h
- Step 7: Set / change the sensor parameters/mode in Main_User.c
- Step 8: Run Main.c



The DNA of tech."

Using The Software Code Library

2. PSoC[®] Cypress

- Step 1: Create a new project in PSoC® Creator
- Step 2: Configure the I2C Master component in TopDesign.cysch
- Step 3: Configure the corresponding I2C pins
- Step 4: Configure the CDC component as well as the USB CDC clock for debugging via COM port purpose
- Step 5: Open the "**C**" folder
- Step 6: Add the following files into the PSoC[®] Creator IDE or other supported IDE:
 - Main.c (From the "PSoC Cypress" folder)
 - I2C_Functions.c
 - "Part Name"_Application _Library.c
 - "Part Name"_PS.c
 - "Part Name"_ALS.c
 - "Part Name"_RGB.c
 - typedefinition.h
 - I2C_Functions.h
 - "Part Name"_Application_Library.h
 - "Part Name".h
 - "Part Name"_Prototypes.h
- Step 7: Activate the MCU-specific code in the files I2C_Functions.c, I2C_Functions.h, and "Part Name"_Application_ Library.c by defining #define Cypress in typedefinition.h
- Step 8: Set / change the sensor parameters / mode in Main.c
- Step 9: Run Main.c



The DNA of tech."

Using The Software Code Library

3. Arduino and Arduino-Supported Platforms Like Teensy

- Step 1: Open the "C++" folder
- Step 2: Add the following files into the Arduino "libraries" folder (usually the "Part Name" folder has to be created):
 - I2C_Functions.cpp
 - "Part Name"_Application_Library.cpp
 - "Part Name"_PS.cpp
 - "Part Name"_ALS.cpp
 - "Part Name"_RGB.cpp
 - typedefinition.h
 - I2C_Functions.h
 - "Part Name"_Application_Library.h
 - "Part Name".h
 - "Part Name"_Prototypes.h

For example: C:\Arduino\libraries\"Part Name"

- Step 3: Add the **Main.ino** from the **"Main"** folder into your desired **"project"** folder, which should be also named as **"Main"** and it should be separated from the above **"libraries"** folder
- Step 4: Activate the MCU-specific code in the files I2C_Functions.cpp, I2_Functions.h, and "Part Name"_Application_Library.cpp by defining #define Arduino in typedefinition.h
- Step 5: Activate the code for Arduino platforms by defining **#define wirelib** in **typedefinition.h** and comment the line **#define i2ct3**. On the other hand, activate the code for Teensy platforms by defining **#define i2ct3** in **typedefinition.h** and comment the line **#define wirelib Note**
 - Arduino platforms use Wire.h library while Teensy platforms use i2c_t3.h library
- Step 6: Open the Arduino sketch
- Step 7: Click File > Open
- Step 8: Select the directory to your already created "project" folder, which contains the Main.ino
- Step 9: Set / change the sensor parameters / mode in Main.ino
- Step 10: Run Main.ino



The DNA of tech."

Using The Software Code Library

4. Other MCUs

- Step 1: Create a new project in your selected IDE
- Step 2: Configure the required pinouts, the I²C components as well as the USB CDC components
- Step 3: Open the "C" folder / "C++" folder
- Step 4: Add the following files into the IDE:
 - I2C_Functions.c/cpp
 - "Part Name"_Application _Library.c/cpp
 - "Part Name"_PS.c/cpp
 - "Part Name"_ALS.c/cpp
 - "Part Name"_RGB.c/cpp
 - typedefinition.h
 - I2C_Functions.h
 - "Part Name"_Application_Library.h
 - "Part Name".h
 - "Part Name"_Prototypes.h
- Step 5: Add the I2C API code of your MCU into I2C_Functions.c/cpp or I2C_Functions.h within the **#ifdef #endif** identifier statement of your MCU. Please ensure that the restart condition is implemented correctly for the I2C read command
- Step 6: Activate the MCU-specific code in the files I2C_Functions.c/cpp, I2C_Functions.h, and "Part Name"_Application_Library.c/cpp by defining #define (write your MCU name) in typedefinition.h
- Step 7: Port the software code from Main.c/ino (from the "PSoC Cypress" / "Main" folder) or Main_User.c (from the "STM32F" folder) and adapt it in the Main.c/cpp based on your selected MCU environment
- Step 8: Port the software code from **"Part Name"_Application_Library.c/cpp** as well as **"Part Name"_Application_Library.h** and adapt it based on your selected MCU environment. Here, only Print functions are MCU-specific and the rest are not. Therefore, you can directly use the non-Print functions for your application if needed
- Step 9: Set / change the sensor parameters / mode in Main.c/cpp
- Step 10: Run Main.c/cpp



The DNA of tech."

Using The Software Code Library

HOW TO USE THE PROJECT EXAMPLES FILES

1. STM32F

- Step 1: Create the following directory: C:\Software\STM Project Note
 - "STM project is directory-specific. For the project file to be directly executed, the above directory has to be established, where the "Software" and "STM Project" folders have to be created in "C:"
- Step 2: Open the "Project Examples" folder
- Step 3: Open the "STM32F" folder
- Step 4: Copy the "Part Name" folder and paste it into the directory mentioned in Step 1



- Step 5: Go into the "Part Name" folder and then again into the "Part Name" folder
- Step 6: Click the **.cproject** file



The DNA of tech™

Using The Software Code Library

I I I I I I I I I I I I I I I I I I I	1035X01 hare View		
Pin to Quick Copy Pas access Clipbo	A Cut te Copy path i≥ Paste shortcut ard Cut Copy path i≥ Paste shortcut Cut Cut Cut Cut Cut Cut Cut Cut Cut C	me New folder New Verw	operties
← → • ↑ <mark>1</mark>	> This PC > Windows (C:) > Software > STM Pro	oject > VCNL4035X01 > VCNL4	035X01
> 📌 Quick access	Name	Date modified	Туре
	.settings	3/16/2021 2:21 PM	File folder
> 📥 OneDrive	📜 Core	3/16/2021 2:21 PM	File folder
> SharePoint	📜 Debug	3/16/2021 2:21 PM	File folder
	Drivers	3/16/2021 2:21 PM	File folder
> 🧢 This PC	Middlewares	3/16/2021 2:21 PM	File folder
> 🥩 Network	USB DEVICE	3/16/2021 2:21 PM	File folder
	DE .cproject	3/9/2021 3:14 PM	CPROJECT File
	.mxproject	3/9/2021 3:14 PM	MXPROJECT File
	DE .project	2/1/2021 7:02 PM	PROJECT File
	STM32F407VGTX_FLASH.Id	3/9/2021 3:14 PM	LD File
	STM32F407VGTX_RAM.Id	2/1/2021 7:02 PM	LD File
	VCNL4035X01 Debug.launch	3/9/2021 4:52 PM	LAUNCH File
	UCNL4035X01.ioc	3/9/2021 3:14 PM	IOC File
	Fig. 9 - The Project File for the S	TM32F Platform	

Step 7: Select the directory to be the same as in Step 1 and click Launch

IDE STM32CubelDE Launcher	\times
Select a directory as workspace	
STM32CubeIDE uses the workspace directory to store its preferences and development artifacts.	
Workspace: ⁹ C:\Software\STM Project\VCNL4035X01 V Browse	
Use this as the default and do not ask again	
Recent Workspaces	
Launch Cancel	
Fig. 10 - Select the Directory	

Document Number: 80289



The DNA of tech."

Using The Software Code Library

Step 8: Ensure that the following 3 lines exist in Main.c:

IDE VCNL4035X01 - VCNL4035X01/Co	ore/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate	e Search Project Run Window Help
🖻 🕶 📓 🖏 💌 🗞 💌 📸 🔍 🕪 💷	■ 23 (3) (3) (2) (三) (3) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ▼ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (2) ♥ (
🏷 Project Explorer 🛛 👘 🗖	🖻 main.c 🖾
🖻 😘 🖓 🖇	210/*
VCNL4035X01	22 * main.c
> 🖑 Binaries	23 *
> 🔊 Includes	24 * Created : 9 November 2020
✓ [™] Core	25 * Modified : 10 March 2021
> 😕 Inc	27 * Version : 5
Y 🗁 Src	28 */
> R I2C Functions.c	29
> Main User c	300/* USER CODE END Header */
	31 /* Includes */
> B stm22f4w hal men c	32 #include "ush device h"
> Sunszi4xz_hai_hsp.c	34
Sunszi4xx_i.c	350/* Private includes*/
> I syscalls.c	36 /* USER CODE BEGIN Includes */
> @ sysmem.c	37 #include "Main_User.h"
> system_stm32f4xx.c	38
> VCNL4035X01_ALS.c	39 /* USER CODE END Includes */



VCNL4035X01 - VCNL4035X01/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

📑 🕶 📾 👒 🕸 🕶 🔦 🕶 📾 i 🗙 🕨 💷	■ 24 3. (> .2) = 〒 〒 (0) @ ▼ 63 ▼ @ ▼ 67 ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ▼ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥ (0) ♥
Project Explorer 🛛 📃 🗮 🗘 🔋	le *main.c ⊠
 VCNL4035X01 Sinaries Includes Core Core Core Core Core Main_User.c Main_User.c 	<pre>106 /* USER CODE BEGIN 2 */ 107 INIT_VCNL4035X01(); 108 109 /* USER CODE END 2 */ 110 111 /* Infinite loop */ 112 /* USER CODE BEGIN WHILE */ 113 while (1) 114 { 115 /* USER CODE END WHILE */ 116</pre>
 is tmaine is stm32f4xx_hal_msp.c is stm32f4xx_it.c is syscalls.c is system.c 	117 /* USER CODE BEGIN 3 */ 118 PRINT_VCNL4035X01(); 119 } 120 /* USER CODE END 3 */ 121 }

Fig. 12 - The initialization function INIT_"Part Name"() and the printing function PRINT_ "Part Name"() need to be included in Main.c Note

Main.c will be updated to a default code by the STM32 Cube IDE. The three lines of codes (INIT_"Part Name"(), INIT_"Part Name"(), and PRINT_"Part Name"(), which call the required header file as well as the initialization and printing functions in Main_User.c, have to be written again when Main.c is updated to the default code after relaunch. The provided Main.c file can obe used as an example of where to add this code if a code rewrite happens

Step 9: Run to execute the project

Document Number: 80289 🔳

For technical questions, contact: <u>sensorstechsupport@vishay.com</u> THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT <u>www.vishay.com/doc?91000</u> VISHAY www.vishay.com

Vishay Semiconductors

The DNA of tech."

Using The Software Code Library

2. PSoC[®] Cypress

- Step 1: Open the "Project Examples" folder
- Step 2: Open the "PSoC Cypress" folder and then "Part Name" folder
- Step 3: Click the "Part Name".cywrk to execute the project

📕 🛃 = VC	CNL4035X01			
File Home	Share View			
Pin to Quick copy access	Paste Paste shortcut	Move Copy to* Copy	ename New folder	Properties History
CI	ipboard	Organize	New	Open S
$\leftarrow \ \rightarrow \ \star \ \uparrow$	VCNL4035X01 > F	Project Examples > PSoC C	ypress > VCNL4035X01	
Quick access	Name	^	Date modified	Туре
~	Backup		3/16/2021 6:17 PM	File folder
OneDrive	VCNL4035X0	1.cydsn	3/16/2021 6:17 PM	File folder
SharePoint	CNL4035X0	1.cywrk	12/15/2020 3:06 PN	1 PSoC Creator Works
	VCNL4035X0	1.cywrk.hwanyusof	2/1/2021 10:51 PM	HWANYUSOF File

Fig. 13 - Run the PSoC® Cypress Workspace

Step 4: Run to execute the project

3. Arduino and Arduino-Supported Platforms Like Teensy

- Step 1: Open the "Project Examples" folder
- Step 2: Open the "Arduino" folder
- Step 3: Add the following "Part Name" folder into the Arduino "libraries" folder

File Home S	hare View			
Pin to Quick Copy Pas access	Cut Copy path Paste shortcut	Move Copy to* to* Delete Rename	New item •	Properties Open
Clipbo	ard	Organize	New	Open
		5		
← → ∽ ↑ 🖡	> VCNL4035X01 > F	Project Examples > Arduino		
$\leftarrow \rightarrow \land \uparrow$	> VCNL4035X01 > F	Project Examples > Arduino	Date modified	Туре
← → • ↑ I	 VCNL4035X01 > F Name Main 	Project Examples > Arduino	Date modified 3/16/2021 2:03 PM	Type File folder

Fig. 14 - Copy the Arduino Library Files

Step 4: Open "Part Name" folder and open typedefinition.h



The DNA of tech."

Using The Software Code Library

- Step 5: Activate the code for Arduino platforms by defining #define wirelib in typedefinition.h and comment the line #define i2ct3. On the other hand, activate the code for Teensy platforms by defining #define i2ct3 in typedefinition.h and comment the line #define wirelib Note
 - Arduino platforms use Wire.h library while Teensy platforms use i2c_t3.h library
- Step 6: Open the Arduino Sketch
- Step 7: Click File > Open
- Step 8: Select the directory to the file Main.ino, which is located in the above "Main" folder

💿 sketch_feb15k	o Arduino 1.8.11				
File Edit Sketch	Tools Help				
	1 I				
sketch_feb15t	2				
🛃 Open an Arc	luino sketch				\times
Look in:	📕 Main		~ 0	🔊 😰 🛄 🔻	
2	Name	^		> Date modified	Ţ
	Main.ino			2/1/2021 11:30 PM	11
QUICK access					
Desktop					
Librarian					
Libraries					
This PC					
1					
Network					
	_				
	<				>
	Object name:	Main.ino		~	Open
	Objects of type:	All Files (*.*)		~	Cancel

Fig. 15 - Open the Arduino Project

Step 9: Run Main.ino

If you experience further issues, please contact us via the support E-Mail: sensorstechsupport@vishay.com