SCP220x Development Kits User's Guide

Document Number: SCP220XDKUG
Rev. 1, 11/2013

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

http://www.freescale.com/support

Document Number: SCP220XDKUG
Rev. 1, 11/2013

Table of Contents

List of Figures

List of Tables

1 Introduction

CogniVue provides several development kits to help with the creation of products with CogniVue CV220x chips in consumer applications and Freescale SCP2200 series chips in automotive applications.

1.1 Development Kits

CogniVue currently provides 3 different development kits, that are development platforms based on CV220x chips: RDK, Automotive PDK and Consumer PDK.

The RDK is a development platform designed around the CV2202 with 128 MB SDRAM.

The PDKs are development platforms designed around the CV2201, with 16 MB stacked memory, and could optionally be equipped with a CV2207, with 64 MB stacked memory (contact CogniVue for details and availability).

All the development kits have programmable permanent memory to host user code for demonstrating user application starting from power up as running on target hardware.

They all have JTAG debug connector for faster development and debugging. Note: a separate JTAG debugger is necessary for this purpose. It is not provided with any of the development kits.

2 Development Kits

2.1 Reference Design Kit (RDK)

The RDK is a full embedded development platform for the CV2202 with 128 MB of SDRAM, an LCD screen and SDHC card slot. The RDK comes standard with a camera module using an OV10633 mega-pixel CMOS sensor.

2.1.1 Overview

The CV2202 RDK features include:

- Main board dimensions of 135mm x 88mm x 18mm (LxWxH), without sensor module attached
- 128MByte DDR SDRAM
- 128Mbyte NAND Flash Memory
- 4.8” WVGA LCD (800x480)
- Audio Codec (WM8990)
- Real Time Clock (DS1339)
- Li-Ion charger (MAX8677C)
- Touchscreen controller (TSC2007)
- Extension slots to interface custom hardware
- Interfaces include:
    - MicroSD/SDHC card slot
    - MiniUSB 2.0 connector
    - 5V DC in connector
    - RCA composite TVOut (NTSC/PAL) connector
    - Mini (3.5mm) Stereo audio out connector
    - TTL UART Tx/Rx
- Interchangeable camera module

2.1.2 Hardware architecture

The hardware block diagram for the CV2202 reference design platform is shown below.

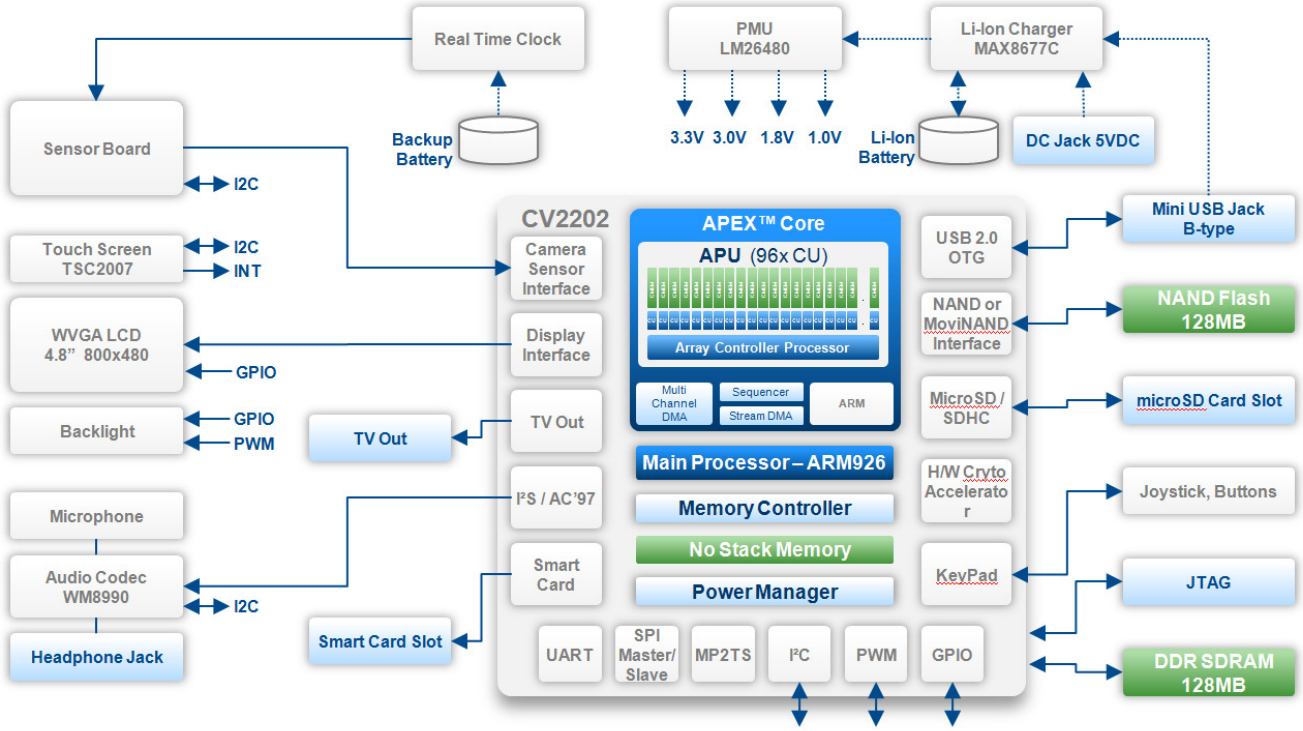


Figure 1: CV2202 reference design

The following pictures illustrate the external physical features of the RDK.

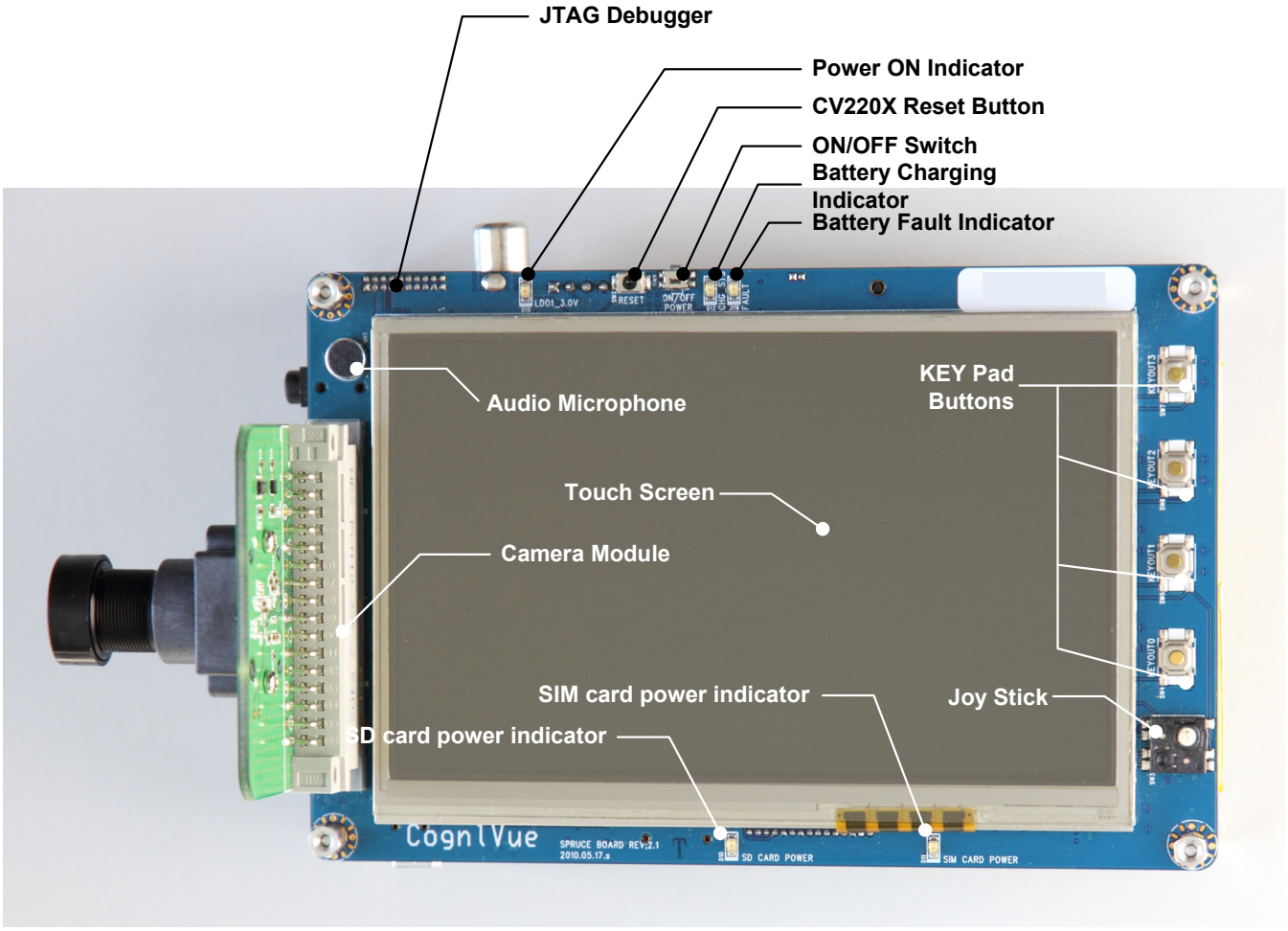


Figure 2: Reference platform top view

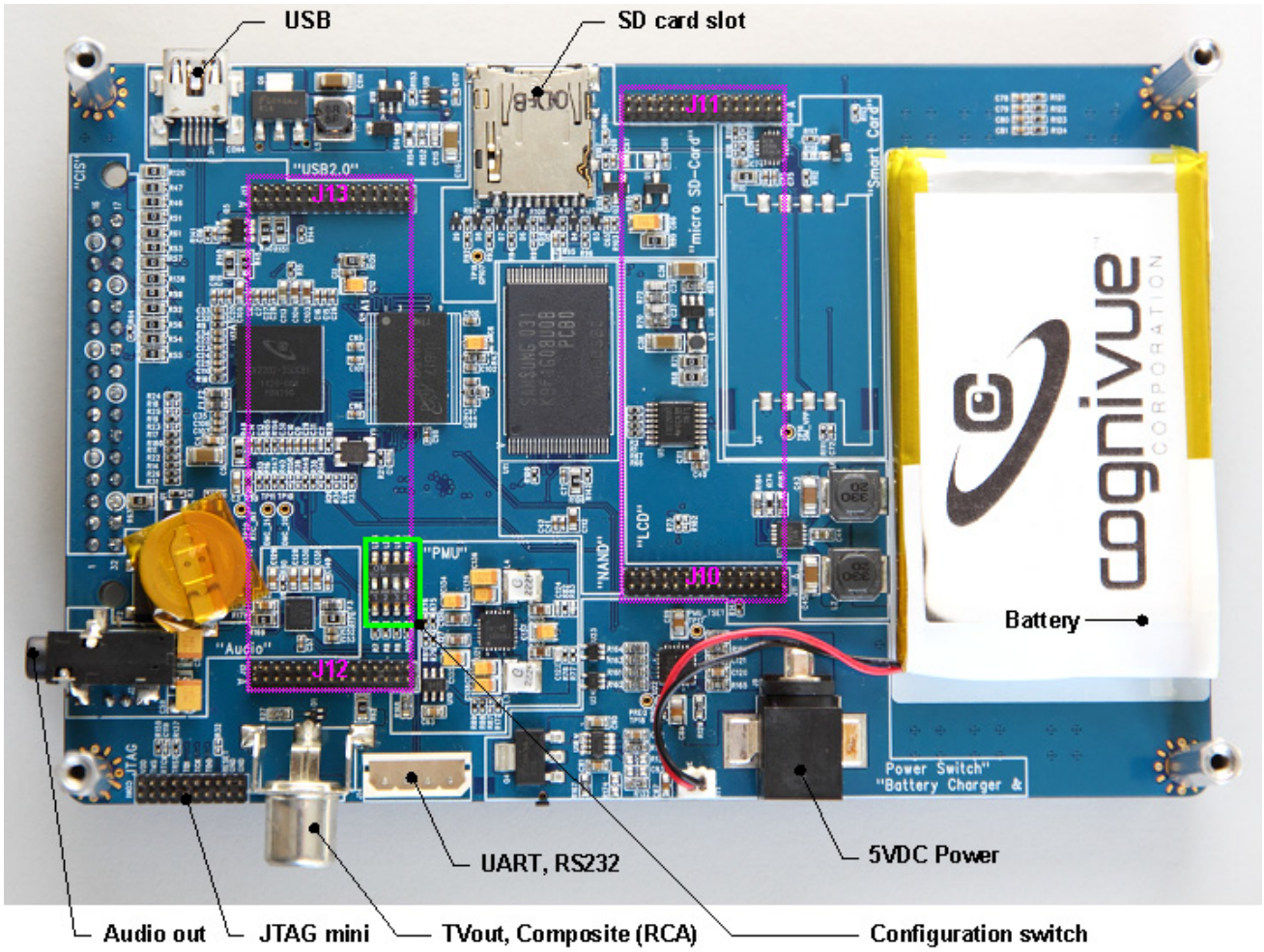


Figure 3: Reference platform bottom view

2.1.2.1 Power supply

The reference platform can be powered by:

- +5V DC adaptor (minimum 1A capacity required for supply)
- USB powered (via USB cable)
- On-board Li-ion battery (1350 mAh)

A MAX8677 Li-ion charger automatically manages the board power sources. With an external source (DC and/or USB) and a battery connected, when the system load is less than the input current limit the battery is charged with the residual power from the input. Battery charging can be disabled with a CV2202 GPIO (GPIO71). Also, if the system load requirements exceed the input current limit, the battery supplies the supplemental current to the load. If no external power is connected, the battery supplies all the system power.

The DC adapter current limit is set to 1A.

The USB current limit is controlled by a CV2202 GPIO (GPIO9). When the GPIO is low, the current limit is 100mA and when the GPIO is high the current limit is 500mA. The USB power source can also be disabled using the USB suspend (GPIO6).

Three LEDs indicate system power status:

- A green LED (LDO1_3.0V) is lit when power is turned ON.
- A green LED (CHG_STS) is lit when the battery is being charged.
- A red LED (FAULT) is lit when a battery charging fault is detected.

As well, several GPIOs also indicate power status.

- CHG_STS is asserted LOW when the battery is being charged (GPIO47).
- FAULT is asserted LOW when a battery charging fault is detected (GPIO57).
- DC_PWR_OK is asserted LOW when the DC adaptor power is ON (WM8990 GPIO3).
- USB_PWR_OK is asserted LOW when the USB power is ON (WM8990 GPIO4).
- 3.1V_DETECT is asserted LOW when the system voltage is OFF (GPIO8).
- 3.4V_DETECT is asserted LOW when the battery voltage is low (GPIO25).

A power management Unit (LM26480) generates the following supply rails for use on the reference platform:

- 1.1V CV2202 core voltage. This power source supplies power to all CV2202 core power islands (LPVDD, CVDD and PTVDD). A CV2202 GPIO (GPIO24) enables the CVDD core power.
- 1.8V CV2202 embedded DRAM voltage
- 3.0V CV2202 IO voltage and peripheral power

- 3.3V CV2202 USB and TV DAC voltage. This power source is enabled using an CV2202 GPIO (GPIO24).

The reference platform has an ON/OFF switch (SW8) which turns off all the power supplies and it also has a RESET button (SW2) that resets the CV2202.

2.1.2.2 Clocks

A single crystal supplies a 24 MHz clock to the CV2202. All other clocks are internally generated in the CV2202 and supplied to the peripherals.

2.1.2.3 Real Time Clock (RTC)

An RTC is available for applications that require real time and/or date stamping. The Maxim DS1339 is a low power clock/date device that is connected to the CV2202 I2C interface and has the capability of running off the battery when the development board is turned OFF.

If the RTC is properly shut down (ie. EOSC=1), then the RTC contents will last for more than 1 year.

2.1.2.4 NAND flash memory

The reference platform contains 128 MB of Nand flash memory (K9F1G08R0B). The Nand flash memory is used for software code storage as well as image and video storage.

The Nand RY/BY busy indicator is connected to a CV2202 GPIO (GPIO72).

2.1.2.5 DDR SDRAM

The reference platform contains 128 MB of DDR SDRAM (MT46H32M32LF) and contains 8 Meg × 32 × 4 banks.

2.1.2.6 USB

The reference platform contains a single USB port utilizing a B-style mini USB connector that supports USB 2.0 OTG.

If the interface is configured as a host (OTG), an LM2731 is enabled to supply the USB VBUS. Otherwise VBUS is an input.

2.1.2.7 Audio

The reference platform uses a WM8990 audio codec.

A stereo audio jack (J2) is connected to the headphone LOUT/ROUT of the codec. As well, a microphone (MK1) is connected to the RIN1 of the codec.

The control path is configured as a 2 wire I2C interface and is connected to the CV2202.

The digital audio interface is I2S. The WM8990 must be configured as a slave on this interface as the transmit and receive frame clock (LRC) are connected together.

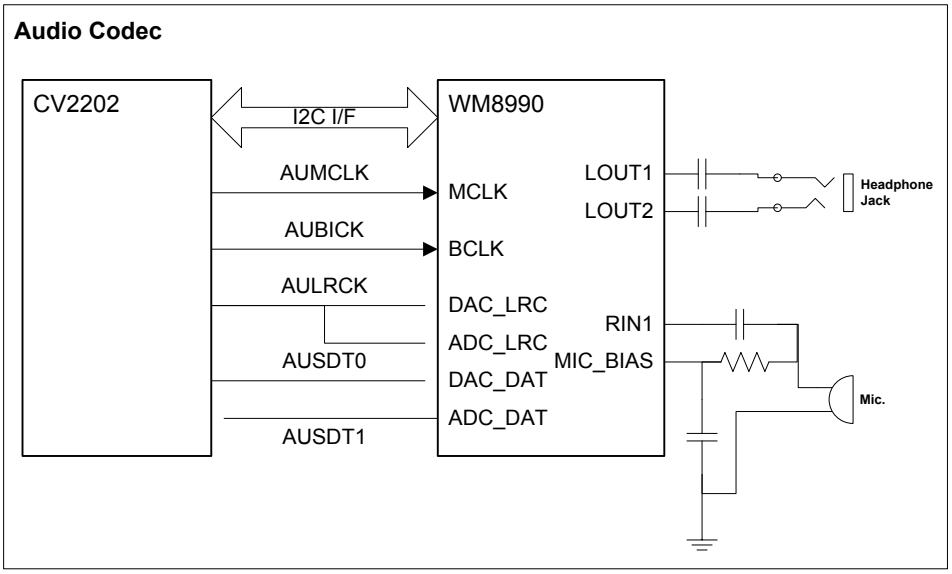


Figure 4: Audio Codec

2.1.2.8 Keypad and joystick

The CV2202 supports a 4 × 4 Key Matrix (16 keys). The reference platform has provided 4 keys and a joystick to allow for menu navigation within the application running on the platform.

2.1.2.9 SD card

The reference platform provides an SD card socket in the TFLASH form factor (microSD). Consult the CV220x data sheet for SD card features.

Several GPIOs are involved with the SD card operation.

- SD_PWC (GPIO49) – SD card power control. Active high enables SD card power.
- SD_CD (GPIO48) – SD card detection. Active low when inserted.

2.1.2.10 Displays (TV and LCD)

The reference platform supports both a TVOut and LCD display.

The TVOut is a composite video that supports both NTSC and PAL and is available on an RCA jack (J1).

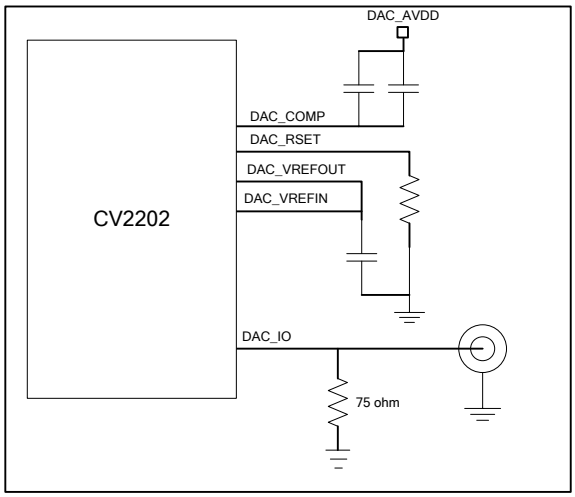


Figure 5: Composite video output

The display module has a touch screen panel on a WVGA LCD (800 × 480).

The touch screen driver is a Texas Instruments TSC2007 and is connected to the CV2202 I2C bus.

The backlight driver is a LT3466 and the brightness is controlled by the CV2202 PWM2.

Several GPIOs are involved with the LCD.

- TSC_IRQ (GPIO46)—Touch screen controller interrupt (active low).
- LCD_PON (GPIO16)—Display power ON.
- LCD_BL_CTRL_PWM (GPIO18)—LCD backlight LED dimming and shutdown control.

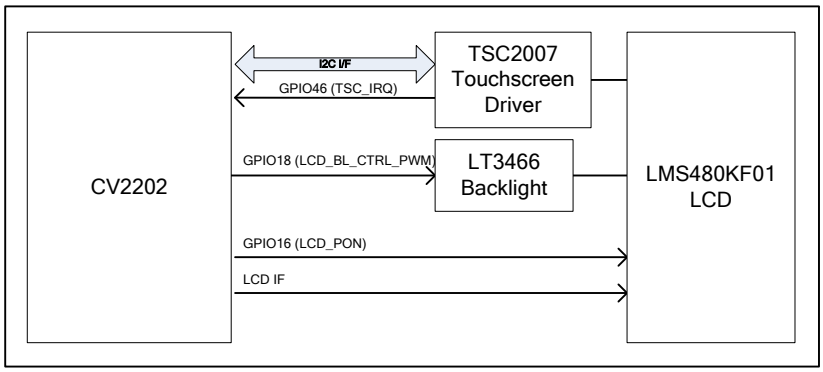


Figure 6: LCD touch screen

2.1.2.11 Sensor

The reference platform supports interchangeable camera modules. The pinout for the module connector is shown in the following table.

Table 1: Camera module pinout

Name	Pin	Pin	Name
D0	1	32	GND
D1	2	31	GND
D2	3	30	CIS_VDD
D3	4	29	VBAT
D4	5	28	
D5	6	27	GND
D6	7	26	CIS_PDN
D7	8	25	SCL
CIS_RST	9	24	SDA
	10	23	GND
	11	22	CIS_PCLK
	12	21	GND
	13	20	CIS_MCLK
	14	19	CIS_VSYNC
D8	15	18	CIS_HREF
D9	16	17	

Two GPIOs are used for the sensor interface.

- CIS_RST (GPIO53)—Sensor Reset.
- CIS_PDN (GPIO52)—Sensor Power Down.

2.1.2.12 RS-232 connector

Connector is a Molex® SPOX™, 22-03-5045.

Table 2: RS-232 connector pinout

J14			
PIN	**Name**	**CV2202 pin number**	**CV2202 pin name**
1	V_BATT (+5VDC)	N.C.	—
2	URX	V18	GPIO50 / UART1RX
3	UTX	V17	GPIO51 / UART1TX
4	Ground	N.C.	—

2.1.2.13 Expansion slots

The reference platform allows for add-on modules through its expansion slots.

- Expansion slot 1: J10 / J11
- Expansion slot 2: J12 / J13

These slots connect to the CV2202 SPI, I2C, UART, and GPIO interfaces.

The following table illustrates slot 1 (J10-J11) connectivity.

Table 3: Expansion slot 1 pinout

J10			
Name	**Pin**	**Pin**	**Name**
VBAT	1	2	VBAT
VBAT	3	4	VBAT
GND	5	6	GND
	7	8	
	9	10	
	11	12	
	13	14	
	15	16	MP2TS_SYNC
GND	17	18	GND
SPI0_CS	19	20	MP2TS_CLK
GND	21	22	GND
SPI0_TX	23	24	MP2TS_DATA
GND	25	26	GND
SPI0_CLK	27	28	MP2TS_VALID
GND	29	30	GND

J11			
Name	**Pin**	**Pin**	**Name**
GND	1	2	GND
	3	4	
	5	6	
	7	8	
	9	10	SCL
GND	11	12	GND
	13	14	SDA
	15	16	
	17	18	
	19	20	
GND	21	22	GND
GPIO2	23	24	
SPI0_RX	25	26	
GPIO0	27	28	
GND	29	30	GND

The following table illustrates slot 2 (J12-J13) connectivity.

Table 4: Expansion slot 2 pinout

J12			
Name	**Pin**	**Pin**	**Name**
VBAT	1	2	VBAT
VBAT	3	4	VBAT
GND	5	6	GND
	7	8	
	9	10	
	11	12	
	13	14	

J13			
Name	**Pin**	**Pin**	**Name**
GND	1	2	GND
	3	4	
	5	6	
	7	8	
	9	10	SCL
GND	11	12	GND
	13	14	SDA

J12			
Name	Pin	Pin	Name
	15	16	
GND	17	18	GND
SPI1_CS	19	20	
GND	21	22	GND
SPI1_TX	23	24	URX
GND	25	26	GND
SPI1_CLK	27	28	UTX
GND	29	30	GND

J13			
Name	Pin	Pin	Name
	15	16	
	17	18	
	19	20	
GND	21	22	GND
GPIO54	23	24	
SPI1_RX	25	26	
GPIO84	27	28	
GND	29	30	GND

2.1.3 Known issues

The following items are known issues with the RDK hardware:

- Occasionally the touchscreen controller stops responding to I2C commands. To resolve this problem the battery must be unplugged from the board (J9) and plugged back in after waiting for a minimum of 5 seconds.
- The RTC backup battery, BT2, can be accidentally pressed against the circuit board preventing proper operation of the RDK. If this happens, inspect the battery and carefully bend it back to position.

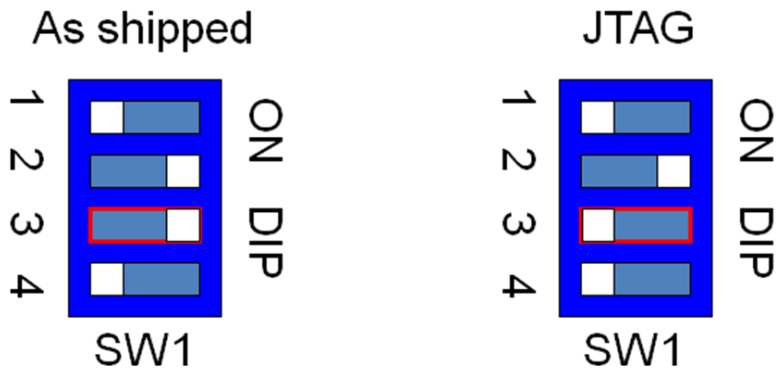
2.1.4 Appendix

2.1.4.1 Power-up Configuration settings (JTAG)

The reference platform has some switch settings (SW1) that control the CV2202 booting method. The following table illustrates the available options.

Table 5: Boot configurations

SW[4]	SW[3]	SW[2]	SW[1]	Boot Mode
NA	OFF	ON	OFF	Boot from SDRAM (code is loaded using the JTAG debugger).
OFF	ON	ON	OFF	Boot from NAND (ECC disabled). Default stand-alone operation
ON	ON	ON	OFF	Boot from NAND (ECC enabled)



2.1.4.2 I2C Slave Addresses

The reference platform has a number of devices connected to the CV2202 I2C bus. The following table provides the slave addresses for these devices.

Table 6: I2C slave addresses

Slave address	Device
1101_000x	DS1339 - RTC
1001_000x	TSC2007—Touch Screen Controller
0011_010x	WM8990—Audio Codec
See code	Sensor Module

2.1.4.3 GPIOs

The following table summarizes the GPIO usage for the CV2202.

Table 7: CV2202 GPIO usage

GPIO	CV2202 direction	Description
GPIO0	input	Optional Module 1 Interrupt
GPIO2	output	Optional Module 1 Power Control (enable HIGH)
GPIO54	output	Optional Module 2 Power Control (enable HIGH)
GPIO84	input	Optional Module 2 Interrupt
GPIO24	output	Core (CVDD) and Analog Power enable
GPIO6	output	USUS—MAX8677 USB Suspend Input
GPIO9	output	USB_CURRENT—MAX8677 USB Current limit control. LOW=100mA, HIGH=500mA.
GPIO47	input	CHARGE_STS—MAX8677 charge status indicator. LOW=battery currently charging.

GPIO	CV2202 direction	Description
GPIO57	input	FAULT—MAX8677 fault indicator. LOW=battery charging error.
GPIO71	output	CHARGE—MAX8677 charge enable. LOW=enable battery charging.
GPIO8	input	3.1V_DETECT—Battery voltage level detection. LOW=battery power is turning OFF.
GPIO25	input	3.4V_DETECT—Battery voltage level detection. LOW=battery power is low.
GPIO16	output	LCD_PON—LCD display ON/OFF.
GPIO18	output	LCD_BL_CTRL_PWM—LT3591 backlight LED dimming and shutoff control.
GPIO46	input	TSC_IRQ—TSC2007 touch screen controller interrupt.
GPIO48	input	SD_CD—SD card detect. LOW=card inserted.
GPIO49	output	SD_PWC – SD card power control. HIGH=power is enabled.
GPIO52	output	CIS_PDN—camera module power down.
GPIO53	output	CIS_RST—camera module reset
GPIO72	input	NF_RB—NAND Flash Ready/Busy indicator.

Because of CV2202 GPIO limitations, some GPIOs are available through the Audio DAC GPIOs.

Table 8: WM8990 GPIO usage

GPIO	CV2202 direction	Description
GPIO3	input	DC_PWR_OK—MAX8667 DC adaptor power OK indicator. LOW=power is OK.
GPIO4	input	USB_PWR_OK—MAX8667 USB power OK indicator. LOW=power is OK.

2.2 Automotive Prototype Design Kit (PDK)

2.2.1 Overview

The Automotive PDK features include:

- HD720p OV10633 Image Sensor*
- 190 degree field of view lens*
- Size: approx 2.5 cm cube, excluding external breakaway PCB extension
- Two GPIOs*
- CV2201-350CBI Image Cognition Processor
- External SPI flash memory—16 MB
- Programmable using CogniVue Image Cognition Processor Software Development Kit (ICP SDK)
- RCA-male composite PAL/NTSC analog video output jack
- 12V DC power
- Module has a breakaway PCB extension with:
    - JTAG connector for debug/development
    - USB2.0 mini connector
    - GPIO connectors
- Typical power consumption ~1W

* PDK Specific. CV2201 ICP supports other configurations

2.2.2 Hardware architecture

The hardware block diagram for the CV2201 Automotive PDK is shown below.

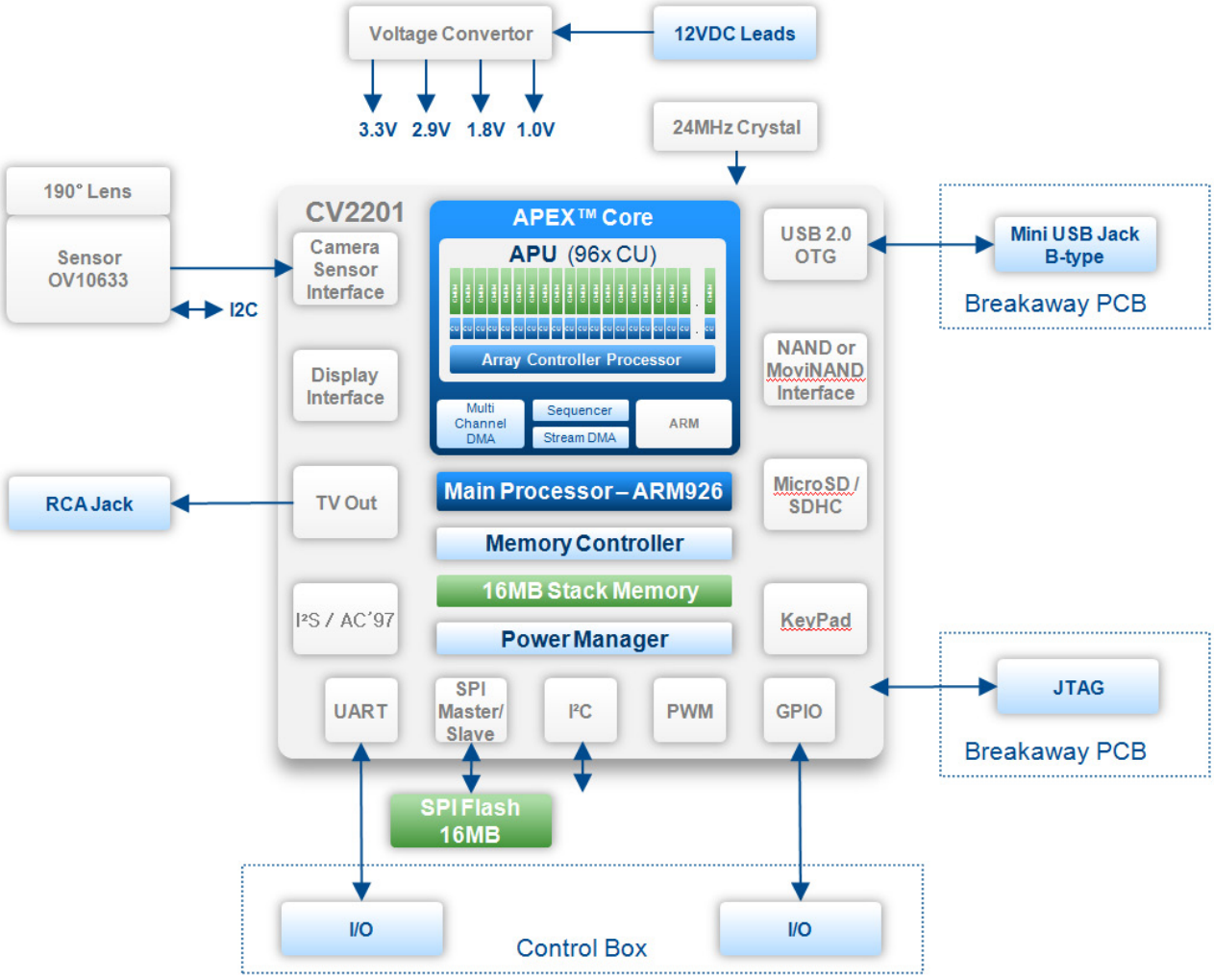


Figure 7: Automotive PDK schematic

The following pictures illustrate the external physical features of the Automotive PDK.

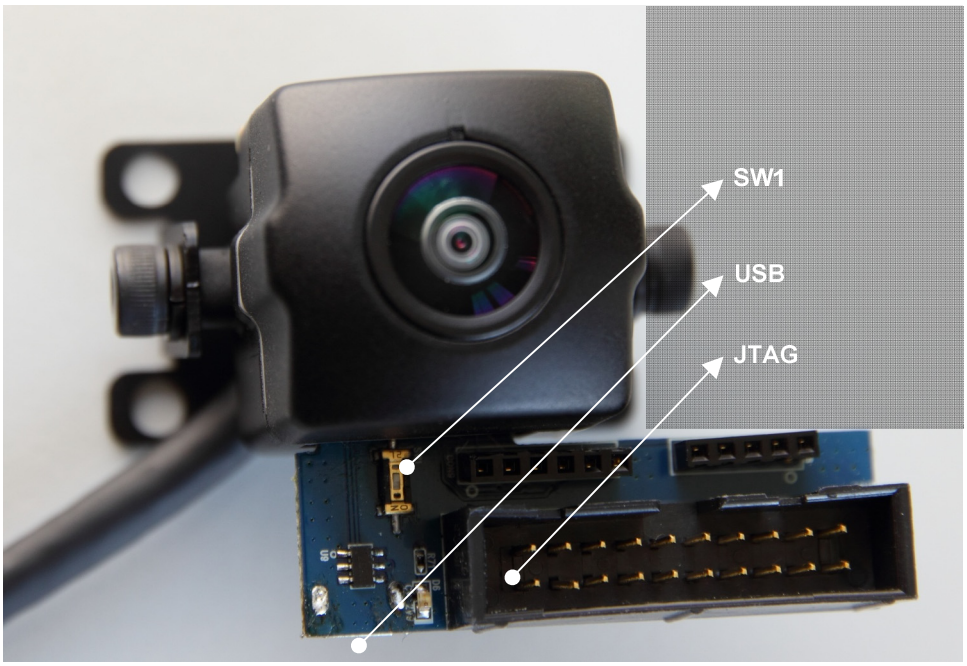


Figure 8: Automotive PDK front view

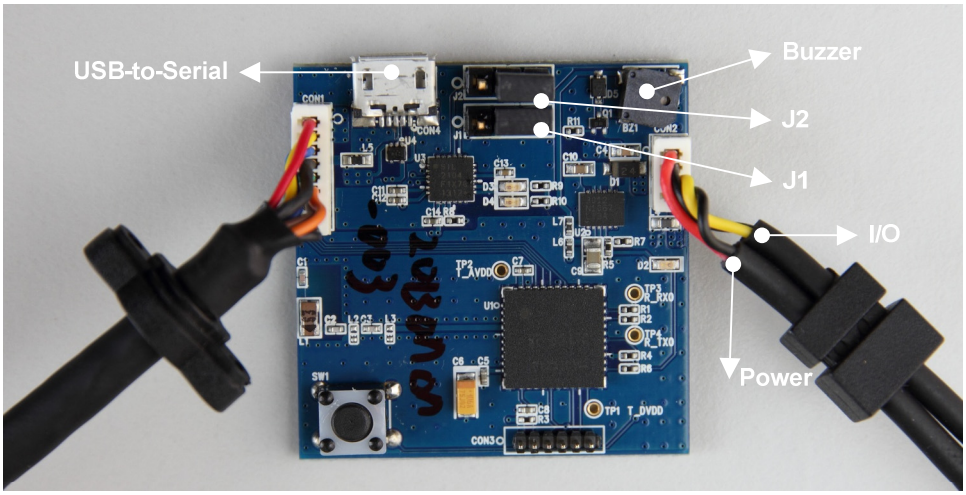


Figure 9: Automotive PDK front view

2.2.2.1 Power supply

The Automotive PDK can be powered by:

- 12V DC power (not provided in kit) through leads (red +12VDC, black Ground)

2.2.2.2 Clocks

A single 24 MHz crystal supplies the master clock for the Automotive PDK.

2.2.2.3 SPI flash

The SPI flash memory in the Automotive PDK is used to hold the program to run at boot up when JTAG is not used.

2.2.2.4 Display

The Automotive PDK has an RCA jack for TV out. It can be NTSC or PAL, selected by software via ICPSDK and is part of the software running on PDK.

2.2.2.5 Sensor

The Automotive PDK integrates an OmniVision® 10633 sensor. Its maximum resolution is 1280 × 800 @ 30fps. It is paired with a 190° lens.

2.2.2.6 Breakaway PCB

The Automotive PDK comes with a breakaway PCB with additional connections for assisting development:

- JTAG connector for debug/development
- Micro USB connector Type-B

It is possible to break away the breakaway PCB but it can’t put back in place afterwards. Also, the connections are then not accessible anymore. It means it can’t be programmed anymore, and the only connection is UART.

2.2.3 Appendix

2.2.3.1 Switch configuration

The consumer PDK has one switch SW1 on breakaway PCB and two jumpers SW1 and SW2 on control box. The following tables illustrate what options are available.

Table 9: SW1 breakaway—JTAG

SW1	JTAG
OFF	Boot from SPI Flash
ON	Boot from JTAG

Table 10: J1—JTAG

J1 & J2	JTAG
1-2	USB-to-UART mode
2-3	Normal mode

2.2.3.2 I/O connectors

The Automotive PDK has few connectors on the control box. They are described as follows:

Table 11: CON1—Control box to camera module

Pin	CON1	CV2201
1	+12VDC in	N.C.
2	EC_TX (output only)	UTX/GPIO22/Pin B1 (**output only**)
3	EC_RX, (input only)	URX/GPIO23/Pin A2 (**input only**)
4	Ground	N.C.
5	XP, (input only)	KEYIN0/GPIO38/Pin G14 (**input only**)
6	XN, (input only)	KEYIN1/GPIO39/Pin G15 (**input only**)
7	YP, (input only)	KEYIN2/GPIO40/Pin G16 (**input only**)
8	YN, (input only)	KEYIN3/GPIO41/Pin F13 (**input only**)

Table 12: CON2—Control box to cable leads

Pin	CON2	CV2201
1	+12VDC in	N.C.
2	EC_TX (output only; if J1 in 2-3, otherwise N.C.)	UTX/GPIO22/Pin B1 (**output only**)
3	EC_RX, (input only; if J2 in 2-3, otherwise N.C.)	URX/GPIO23/Pin A2 (**input only**)
4	Ground	N.C.

2.3 Consumer Prototype Design Kit (PDK)

2.3.1 Overview

The CV2201 Consumer PDK features include:

- HD720p OV10633 Image Sensor*
- 120 degree field of view lens*
- Size: Approx. 3 cm deep × 9 cm wide × 4 cm high, including adjustable mounting stand
- CV2201-350CBI Image Cognition Processor (16 MB SDRAM)
- External SPI flash memory—16 MB
- USB2.0 mini connector
- Programmable using CogniVue Image Cognition Processor Software Development Kit (ICP SDK)
- Module includes a breakaway PCB extension with the following connectors:
    - JTAG connector for debug/development
    - RCA composite PAL/NTSC analog video output connector
    - 5V DC power connector
    - GPIO header connector
- Typical power consumption ~1W

* PDK Specific. CV2201 ICP supports other configurations

2.3.2 Hardware architecture

The following diagram illustrates the elements present in the Consumer PDK.

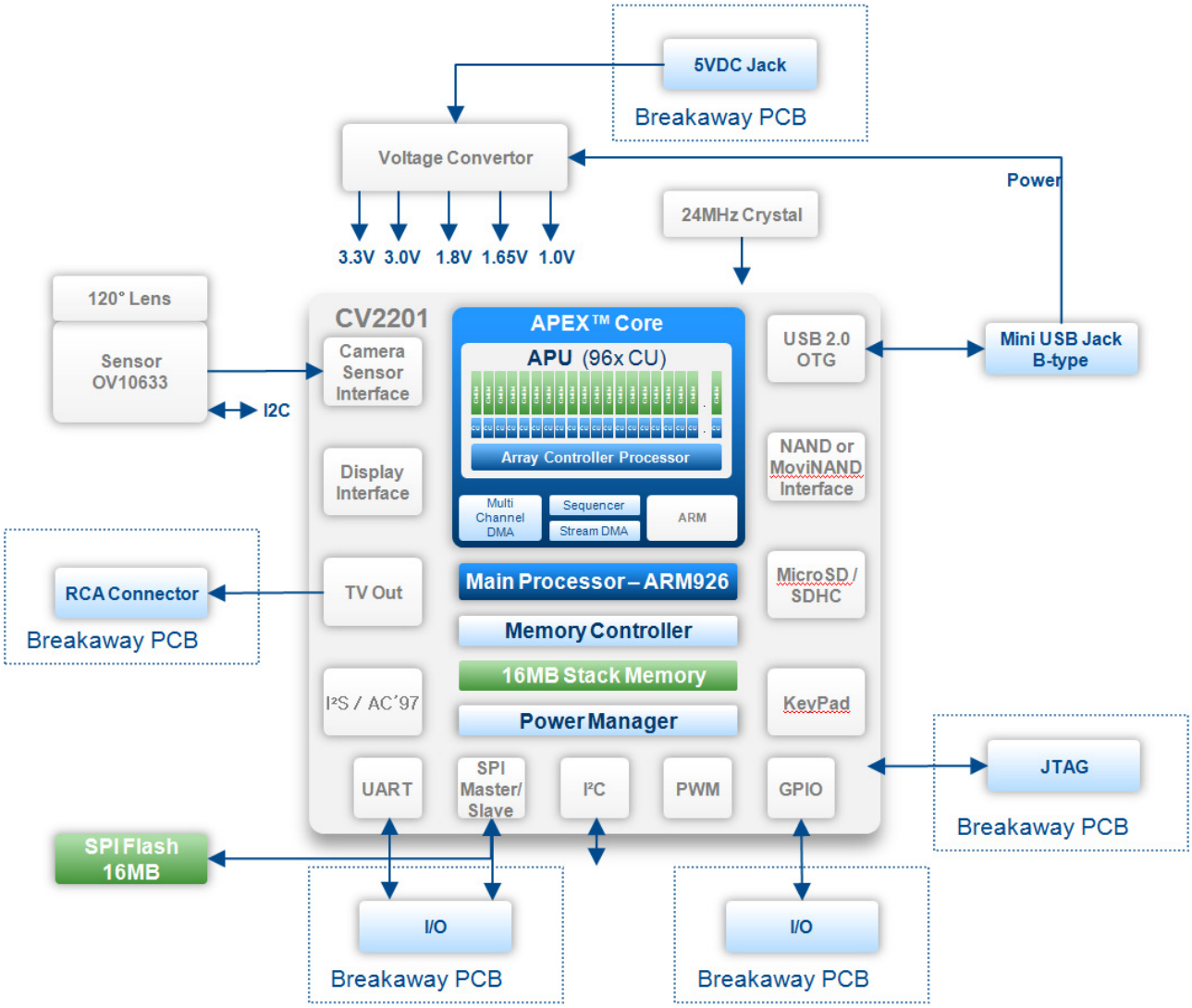


Figure 10: Consumer PDK hardware architecture

The following illustrations indicate the physical location of the consumer PDK features.

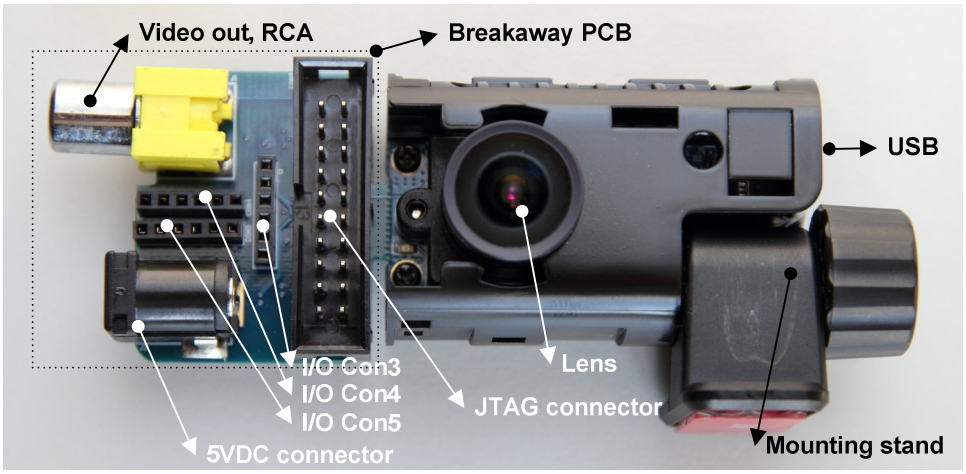


Figure 11: Consumer PDK Front View

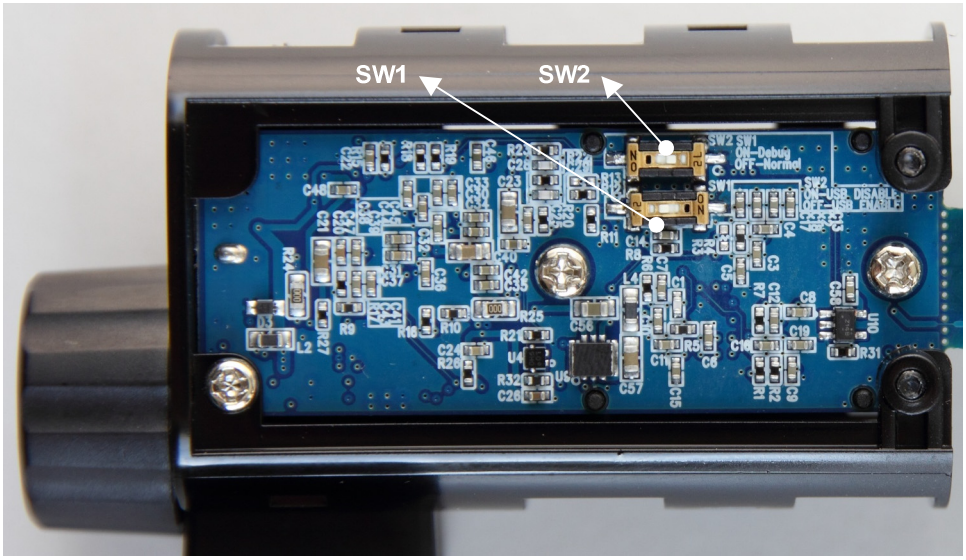


Figure 12: Consumer PDK bottom view

2.3.2.1 Power supply

The Consumer PDK can be powered by:

- USB cable (when connected to a powered USB port)

- 5VDC power adaptor (not provided in kit)

(SW1 must be in OFF position)

2.3.2.2 Clocks

A single 24 MHz crystal supplies the master clock for the consumer PDK.

2.3.2.3 SPI flash

The SPI flash memory in the consumer PDK is used to hold the program to run at boot up when JTAG is not used.

2.3.2.4 Display

The consumer PDK has a RCA connector for TV out. It can be NTSC or PAL, selected by software via ICPSDK and is part of the software running on PDK.

2.3.2.5 Sensor

The consumer PDK integrates an OmniVision 10633 sensor. Its maximum resolution is 1280 × 800 @ 30fps. It is paired with a 120° lens.

2.3.2.6 Breakaway PCB

The consumer PDK comes with a breakaway PCB with additional connections for assisting development:

- JTAG connector for debug/development
- RCA composite PAL/NTSC analog video output connector
- 5V DC power connector
- GPIO header connector

It is possible to break away the breakaway PCB but it can’t put back in place afterwards. Also, the connections are then not accessible anymore. It means it can’t be programmed anymore, and the only connection is USB.

2.3.3 Appendix

2.3.3.1 Switch configuration

The consumer PDK has two switches: SW1 and SW2. The following table illustrates the available options.

Table 13: SW1—Disable USB switch

SW1	Disable USB
OFF	USB is functional
ON	USB is disabled

Table 14: SW2—JTAG

SW2	JTAG
OFF	Boot from SPI Flash
ON	Boot from JTAG

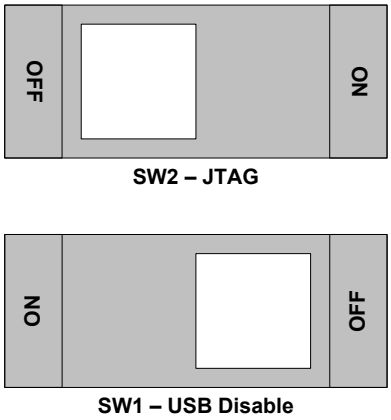


Figure 13: Consumer PDK SW1 and SW2

2.3.3.2 I/O connectors

The Consumer PDK has 3 I/O connectors on its Breakaway PCB. They offer additional connectivity as described in the following tables. Please do not use CON3, usage is reserved.

Table 15: CON3 pinout (Reserved)

CON3 (Reserved)			
PIN	**Name**	**CV2201 pin number**	**CV2201 Pin name**
1	CV_VDDIO	N.C.	—
2	SPI_TX	C4	GPIO26 / SPI0_TX
3	SPI_RX	C5	GPIO27 / SPI0_RX
4	SPI_CS	C3	GPIO28 / SPI0_SSN
5	SPI_CK	C2	GPIO29 / SPI0_SCK
6	Ground	N.C.	—

Table 16: CON4 pinout

CON4			
PIN	**Name**	**CV2201 pin number**	**CV2201 pin name**
1	CV_VDDIO	N.C.	—
2	GPIO41	F13	GPIO41 / KEYIN3
3	GPIO40	G16	GPIO40 / KEYIN2
4	GPIO39	G15	GPIO39 / KEYIN1
5	GPIO38	G14	GPIO38 / KEYIN0
6	Ground	N.C.	—

Table 17: CON5 pinout

CON5			
PIN	**Name**	**CV2201 pin number**	**CV2201 pin name**
1	CV_VDDIO	N.C.	—
2	N.C.	—	—
3	URX	A2	GPIO23 / URX
4	UTX	B1	GPIO22 / UTX
5	N.C.	—	—
6	Ground	N.C.	—

3 Software Development Kit

Development of custom applications for the CV220x is made by cross-compilation from a personal computer with Microsoft® Windows operating system.

The CV220x development kits come with an operating system, libraries and tools to build custom applications: all are provided part of the ICP SDK, Image Cognition Processor Software Development Kit.

This package can be extended with toolkits (at extra cost), such as the APEX toolkit, the Image Dewarping Toolkit, and the Image Processing Algorithm Toolkit.

ICP SDK is available from CogniVue website at: http://www.cognivue.com/kits

Contact CogniVue to receive access to this page content.

To download the ICP SDK package, select the link associated to "CV220x ICP SDK Software." The package is an executable installer for Microsoft Windows; just run and follow the instructions as describe below.

3.1 Package installation

Run CV220X_RDK_ICP-SDK-x.y.z-Setup.exe where x.y.z is the ICP SDK version number.

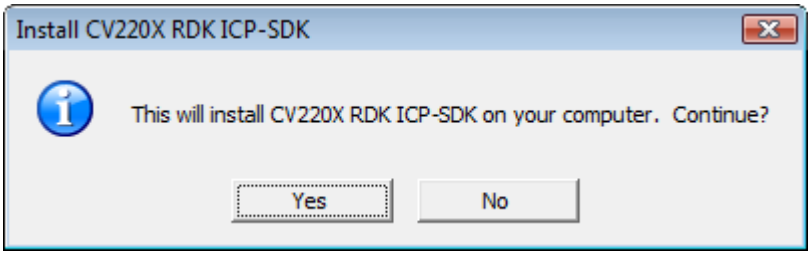


Figure 14: Install ICP SDK step 1

Select 'Yes' to continue.

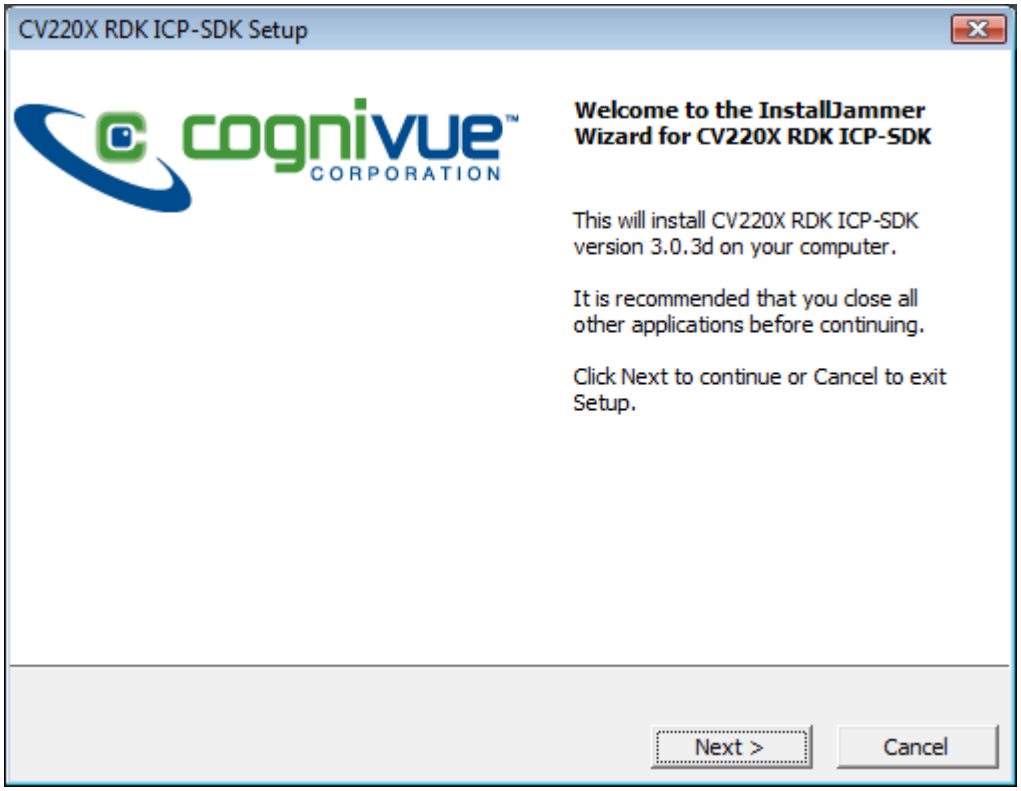


Figure 15: Install ICP SDK step 2

Select ‘Next >’ to continue.

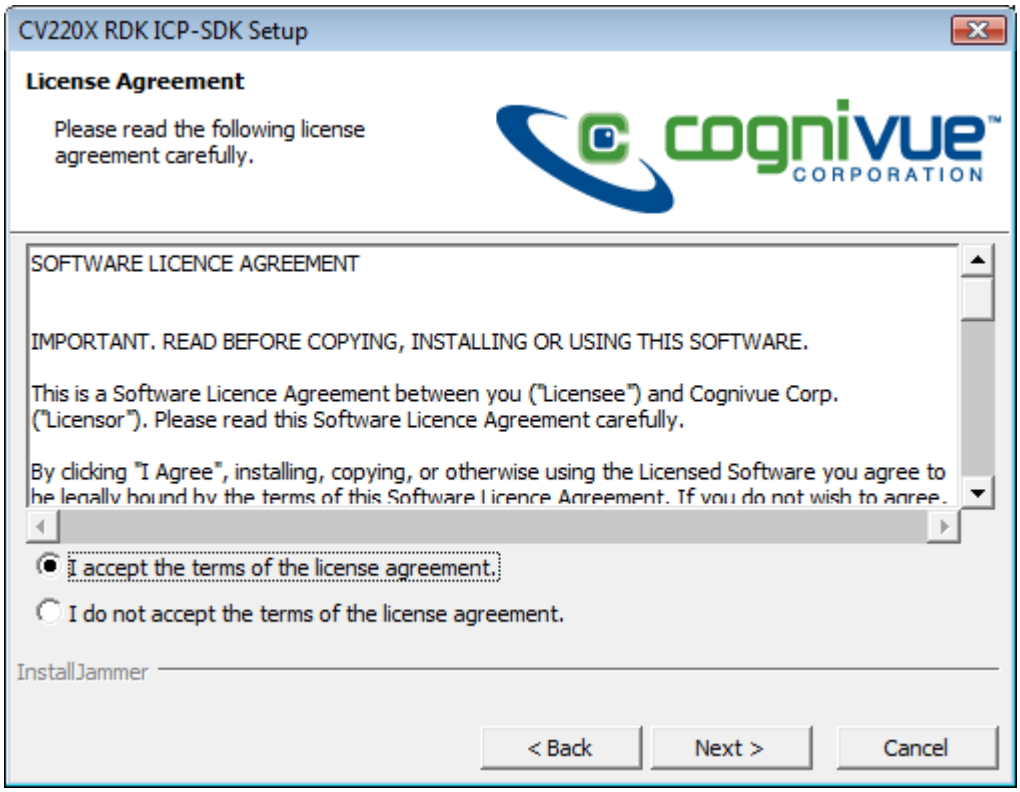


Figure 16: Install ICP SDK step 3

Read and select ‘Next >’ to accept the license agreement and continue with the installation.

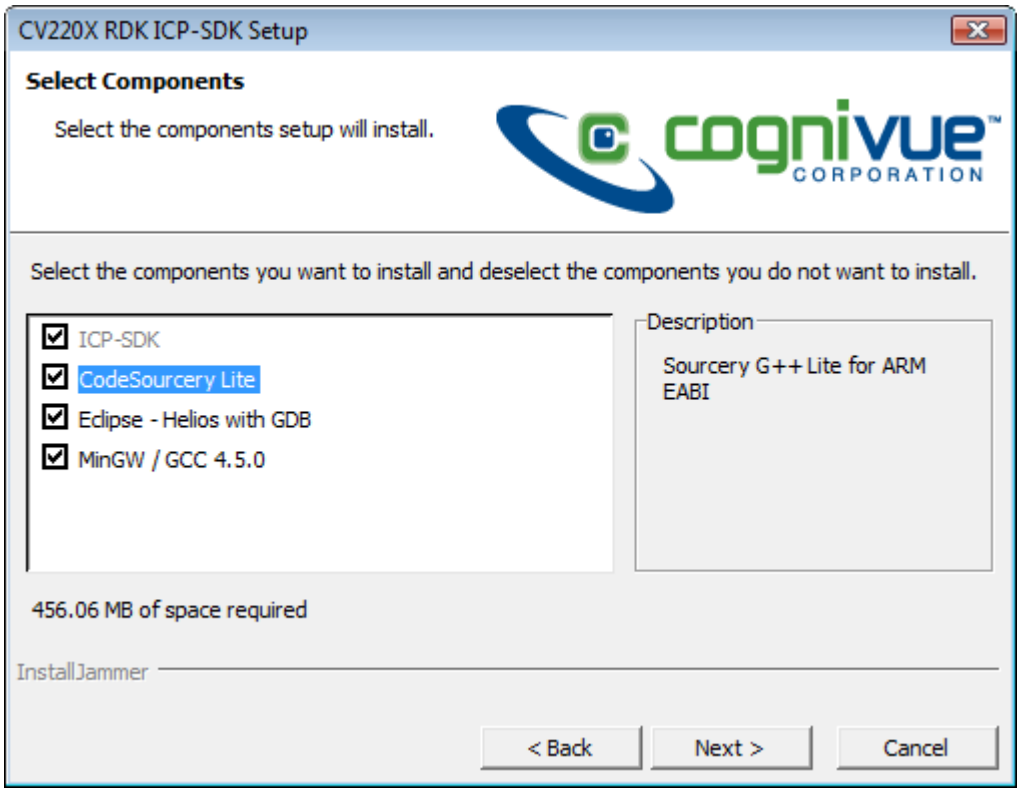


Figure 17: Install ICP SDK step 4

At this step, it is possible to select the components to install:

- ICP SDK is mandatory
- CodeSourcery™ Lite: this is Sourcery G++ Lite, an ARM® GNU toolchain: compiler, assembler, linker (needed for development, uncheck if already installed).
- Eclipse™: As a code editor and IDE platform (optional, it is possible to use your own editor; uncheck if already installed)
- MinGW: compiler and tools use for APEX code generation (needed for development, uncheck if already installed)

Then, select ‘Next >’ to continue.

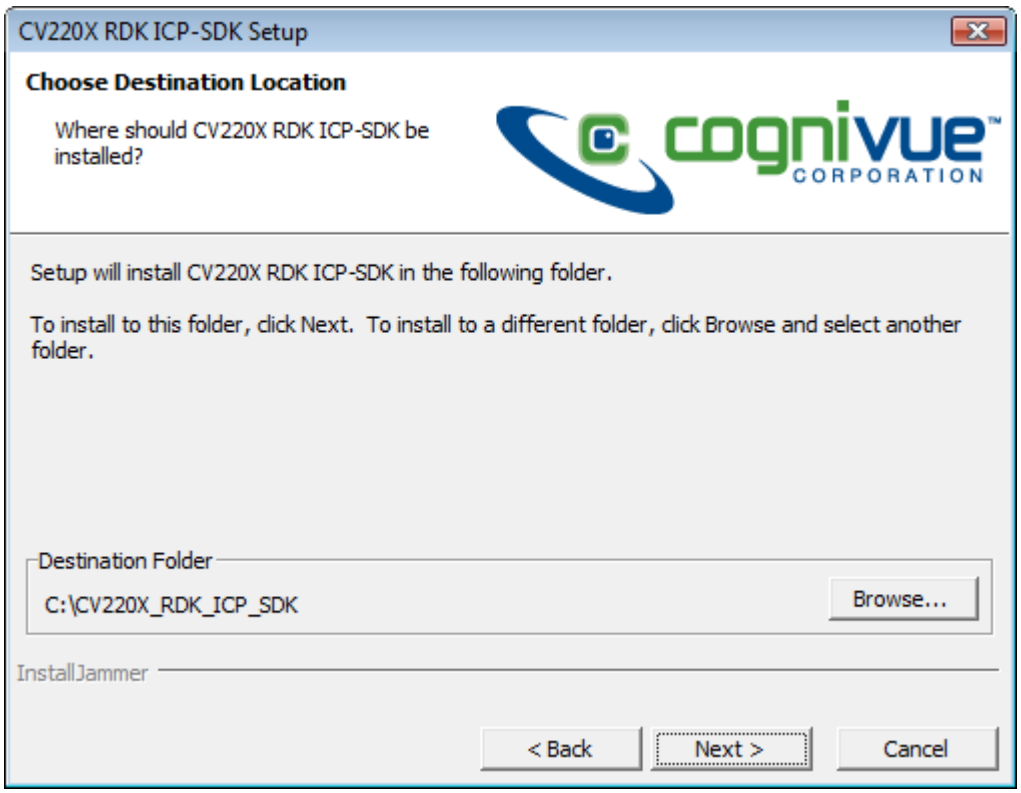


Figure 18: Install ICP SDK step 5

At this step, it is possible to select the destination folder.

Select ‘Browse…’ to change the path.

Then, select ‘Next >’ to continue.

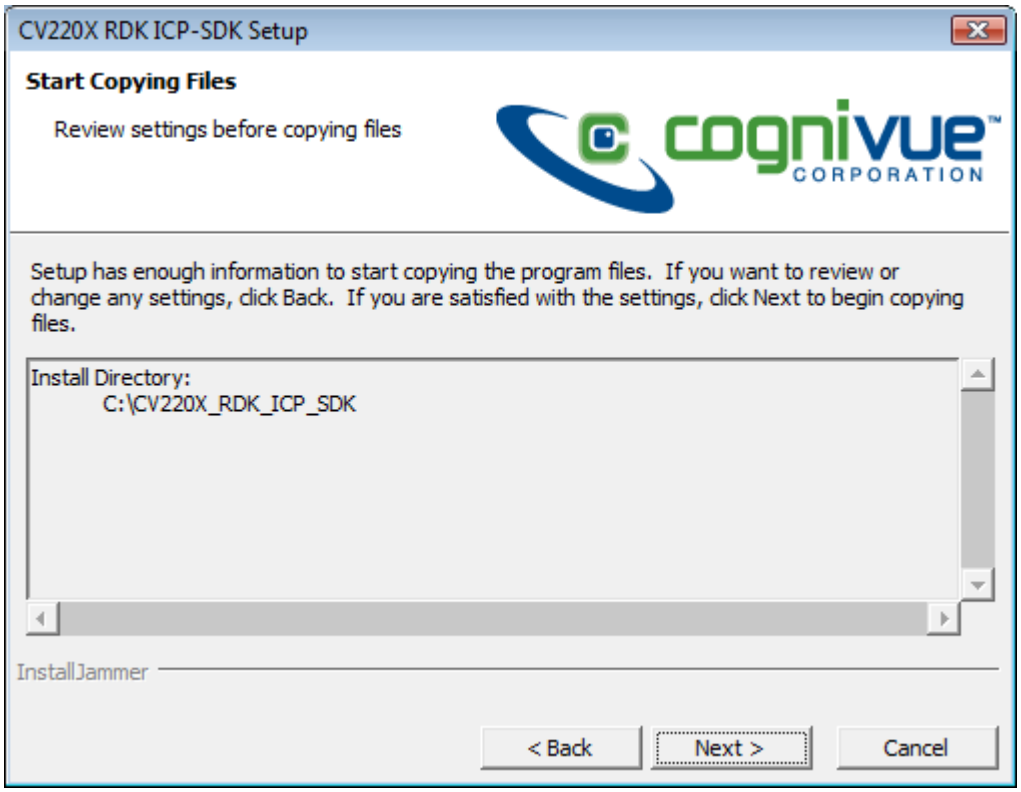


Figure 19: Install ICP SDK step 6

Verify and select ‘Next >’ to continue.

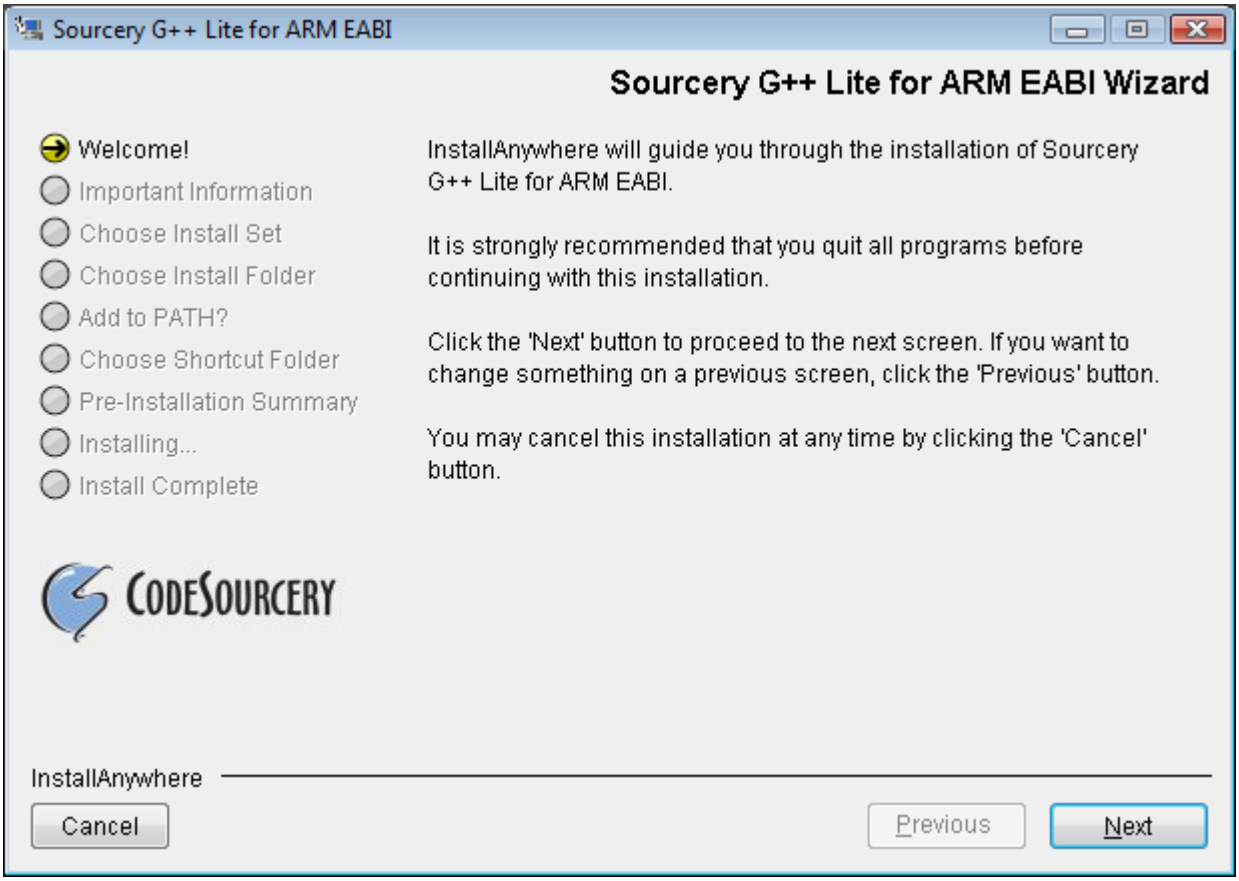


Figure 20: Install ICP SDK step 7

Now, Sourcery G++ Lite is being installed.

Select ‘Next >’ to continue.

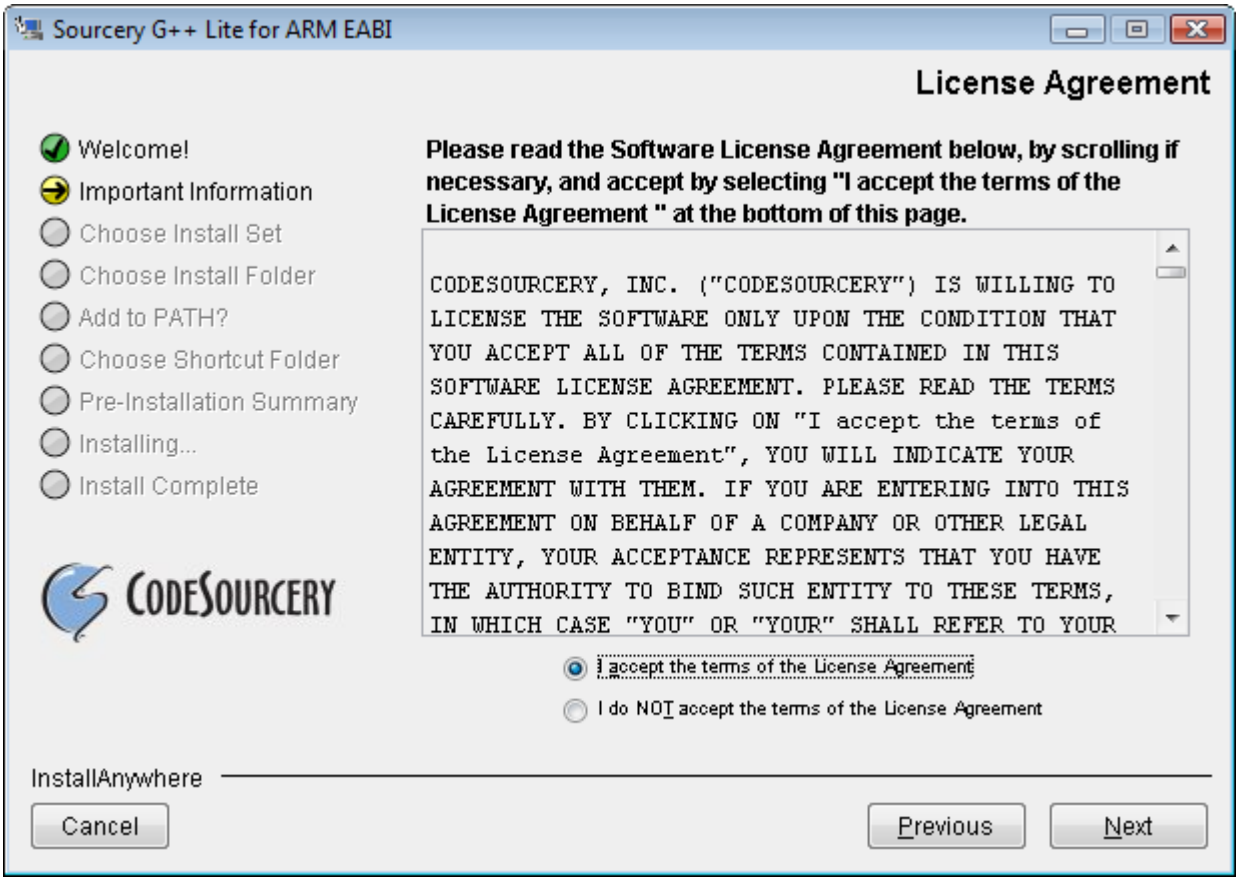


Figure 21: Install ICP SDK step 8

Select ‘Next >’ to continue.

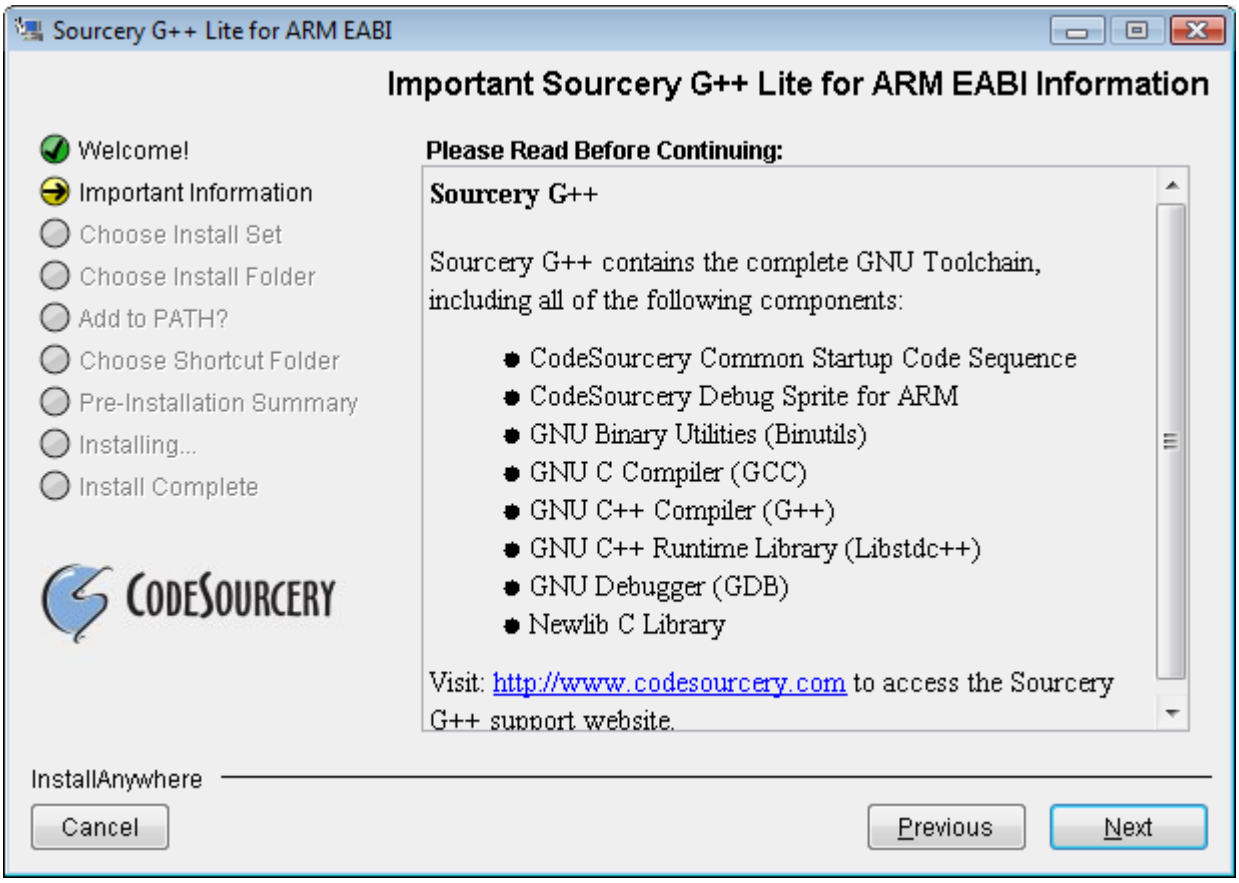


Figure 22: Install ICP SDK step 9

Select ‘Next >’ to continue.

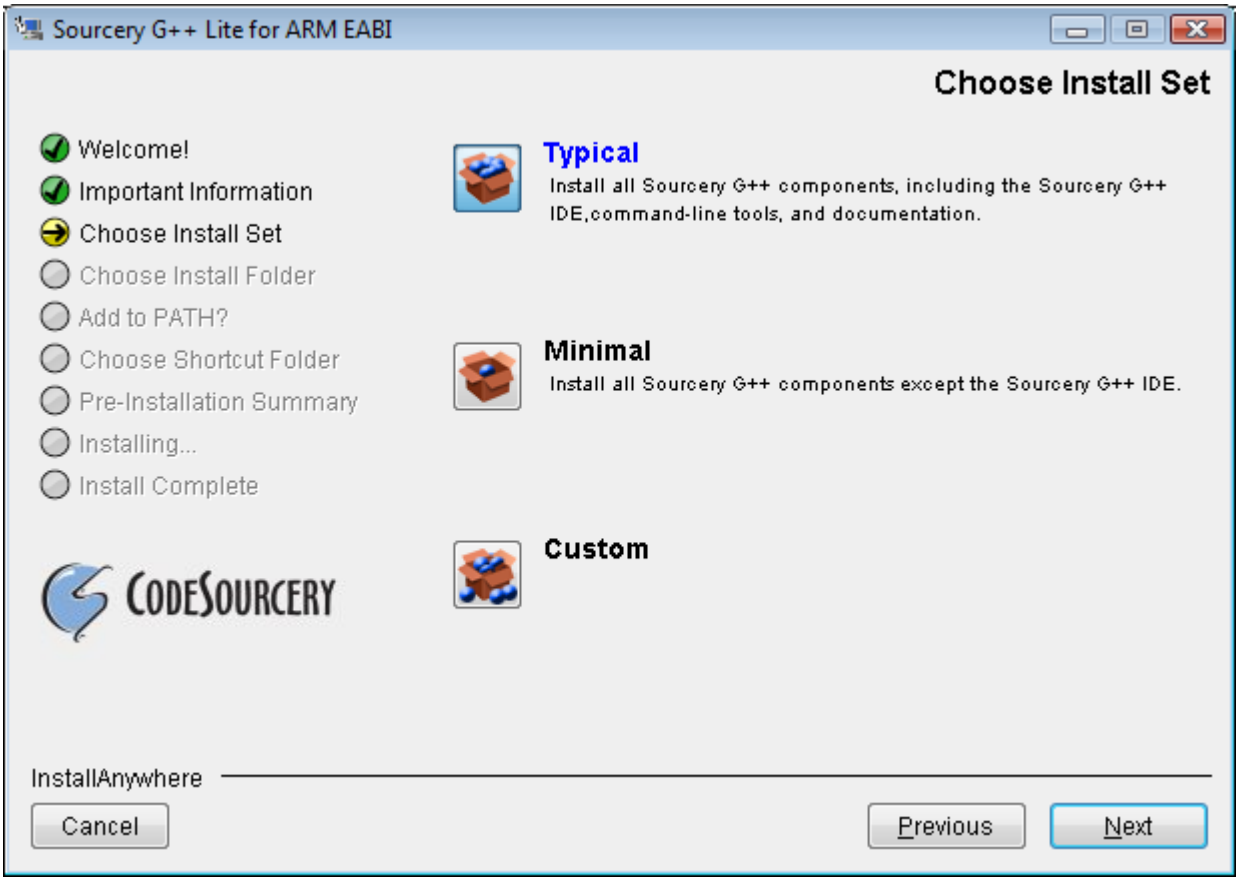


Figure 23: Install ICP SDK step 10

Select ‘Next >’ to continue.

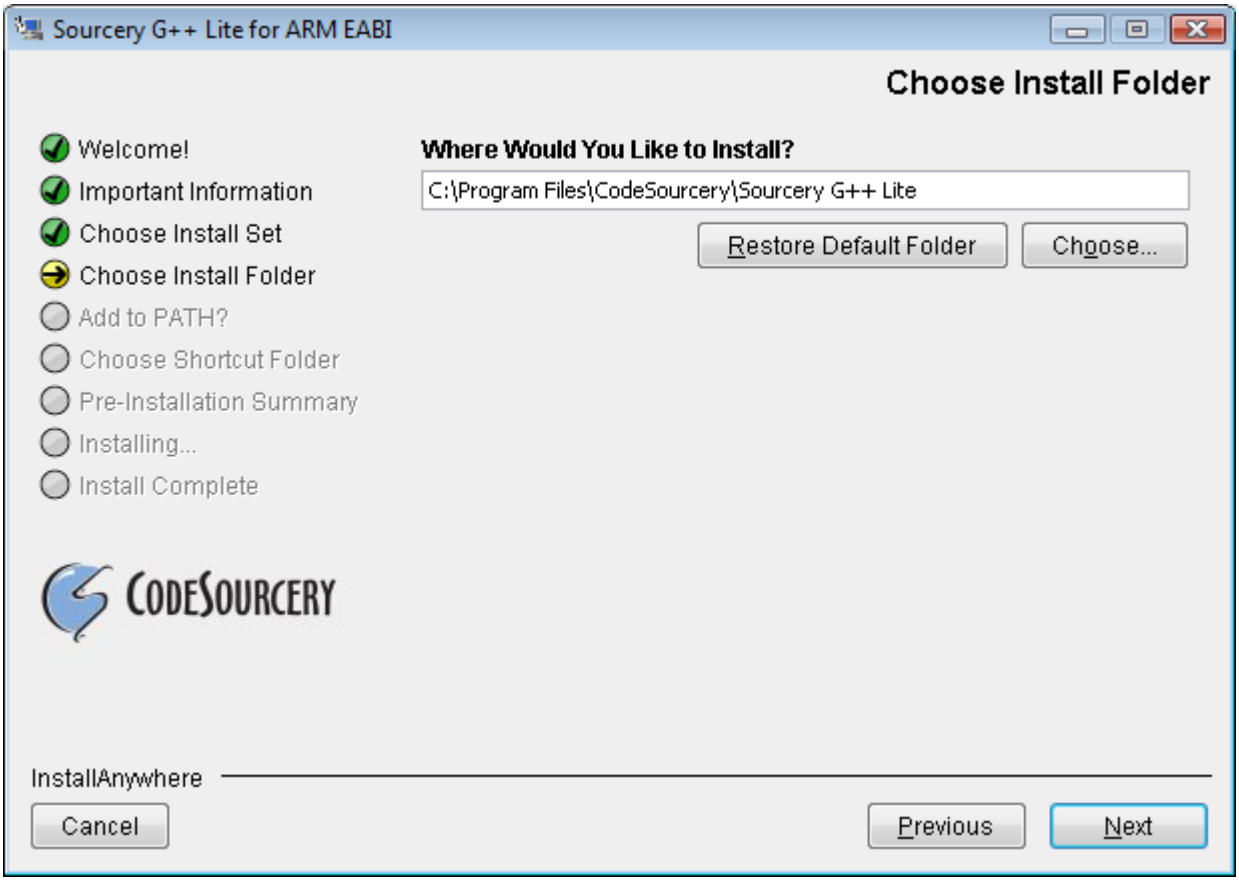


Figure 24: Install ICP SDK step 11

Select ‘Next >’ to continue.

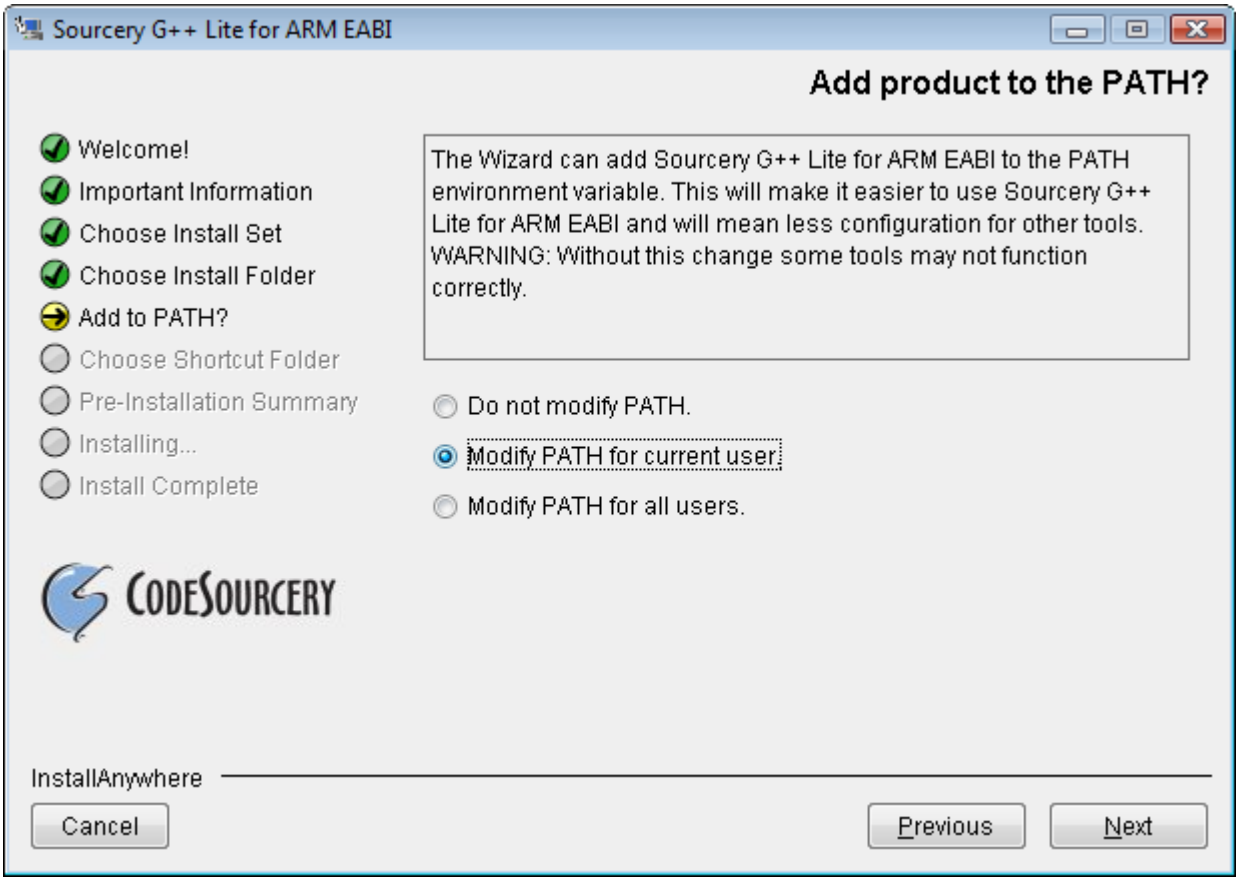


Figure 25: Install ICP SDK step 12

Select ‘Next >’ to continue.

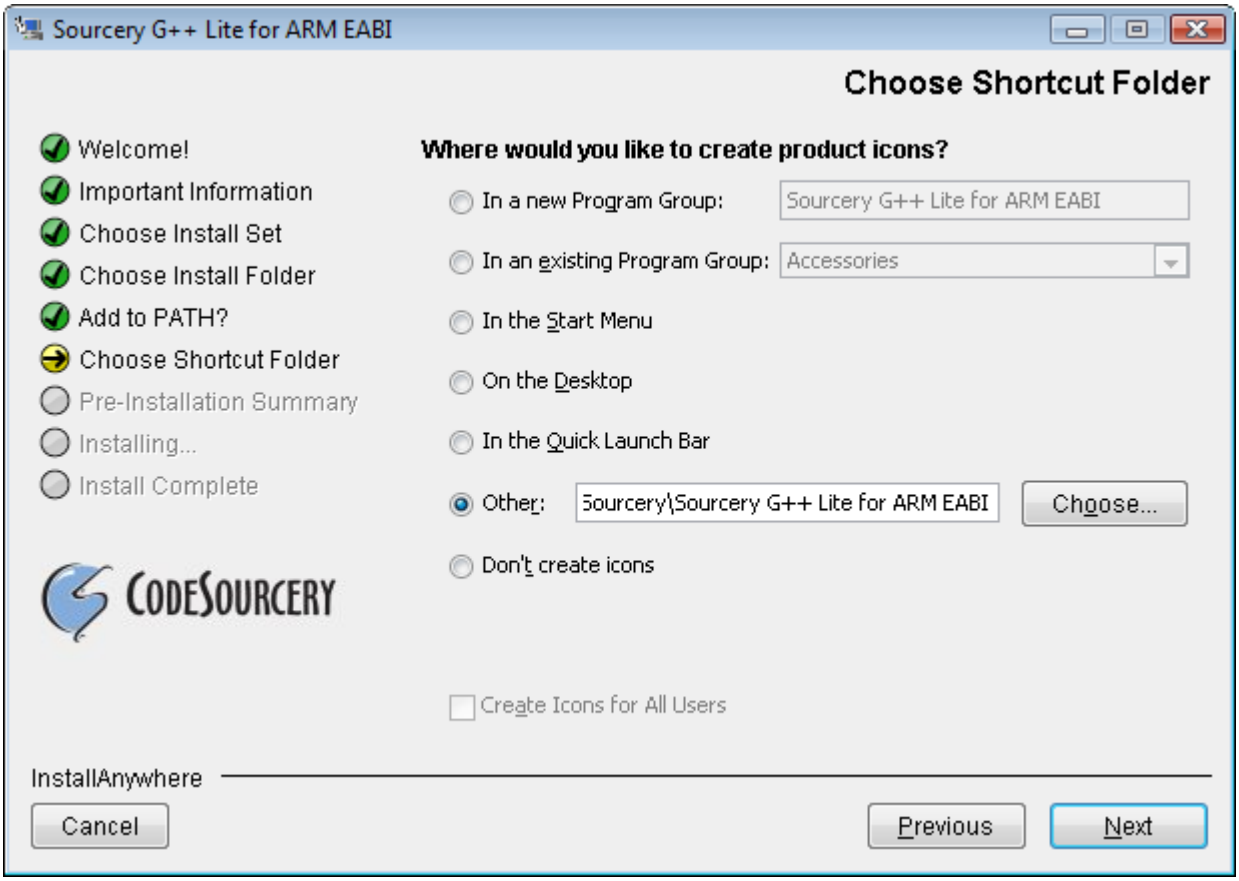


Figure 26: Install ICP SDK step 13

Select ‘Next >’ to continue.

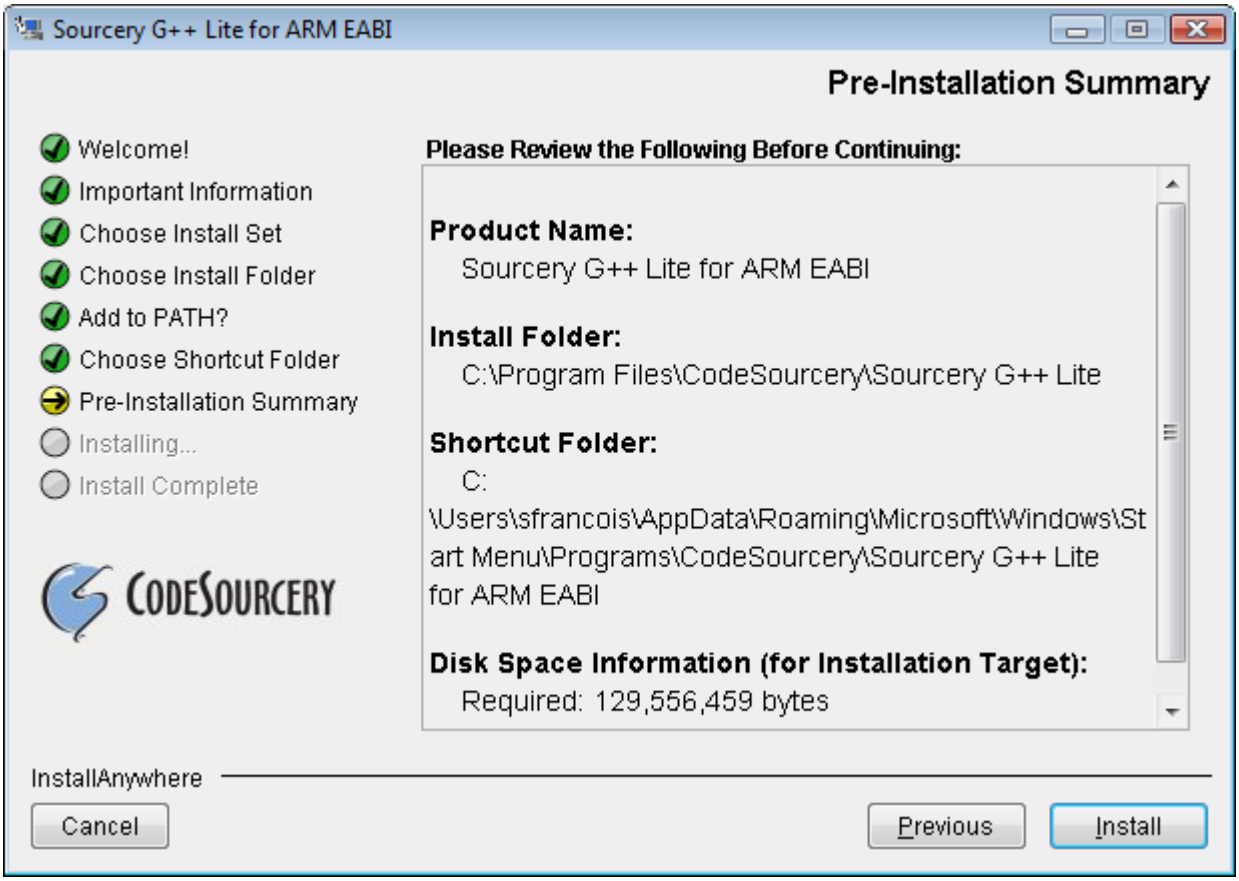


Figure 27: Install ICP SDK step 14

Select ‘Next >’ to continue.

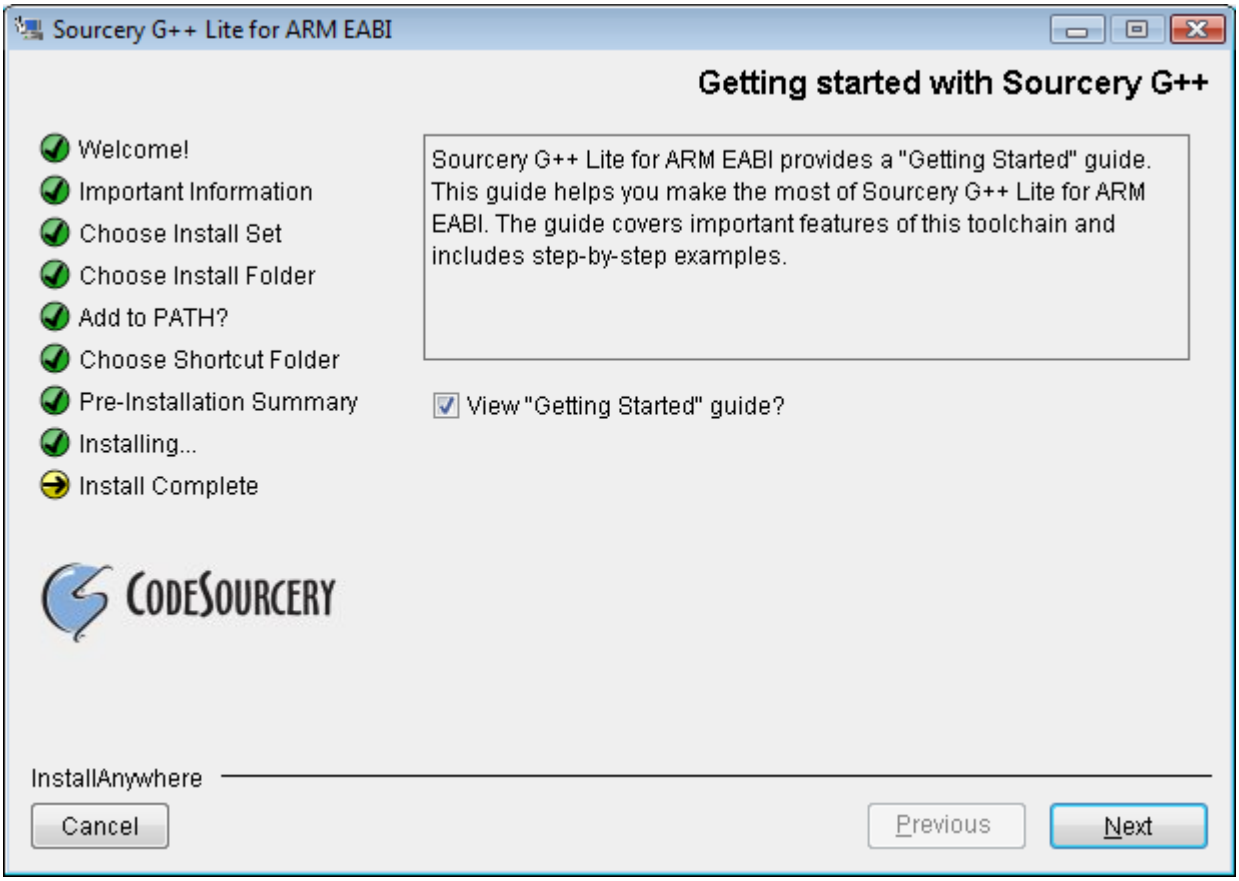


Figure 28: Install ICP SDK step 15

Select ‘Next >’ to continue.

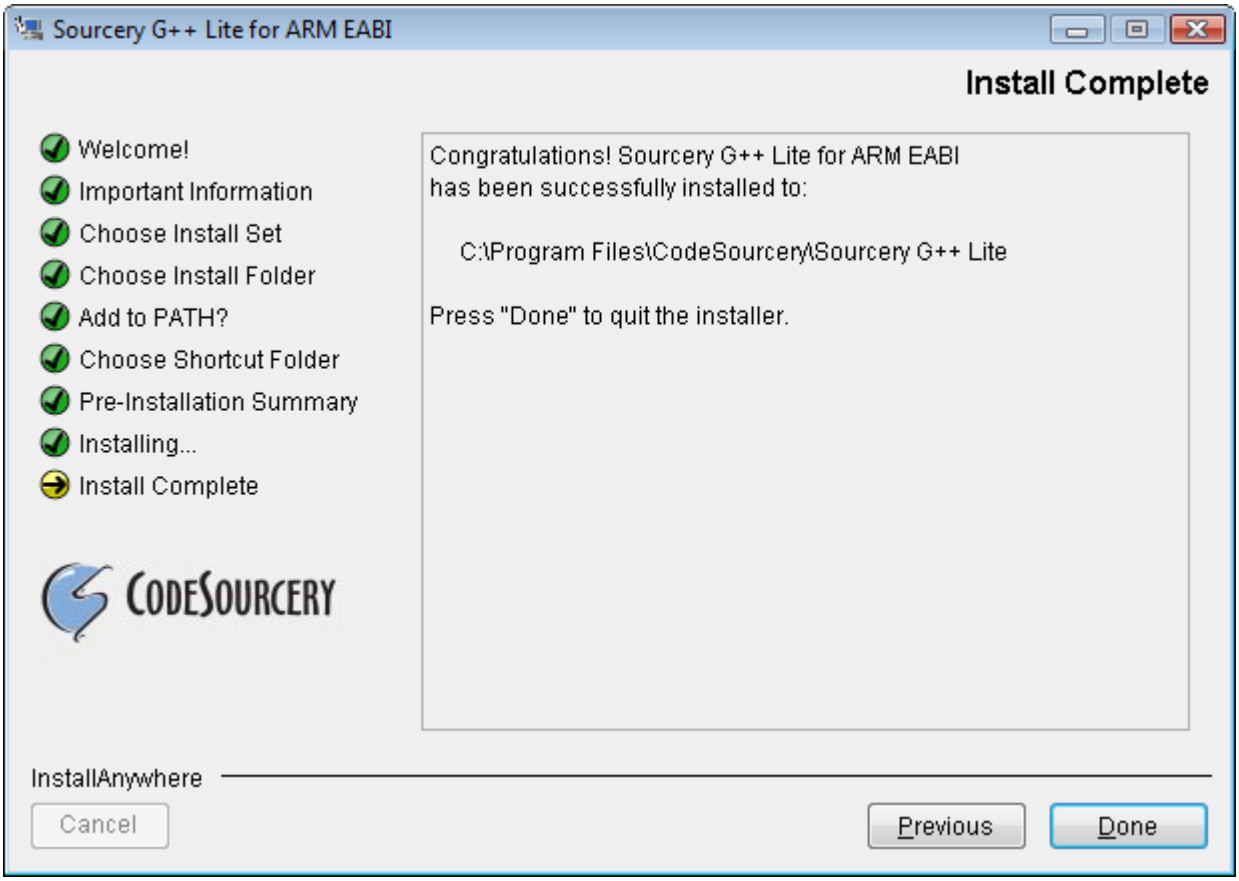


Figure 29: Install ICP SDK step 16

Sourcery G++ Lite has been installed.

Select ‘Done’ to continue.

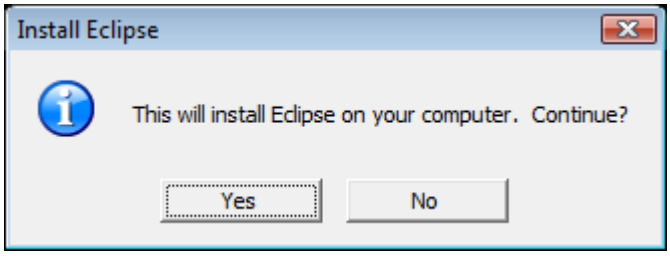


Figure 30: Install ICP SDK step 17

Now, Eclipse is being installed.

Select ‘Yes’ to continue. This is an optional component, this part may be skipped if another code editor will be used. If a debugger is used, it may require Eclipse. If Eclipse is not being installed, go to step Figure 35: Install ICP SDK step 22.

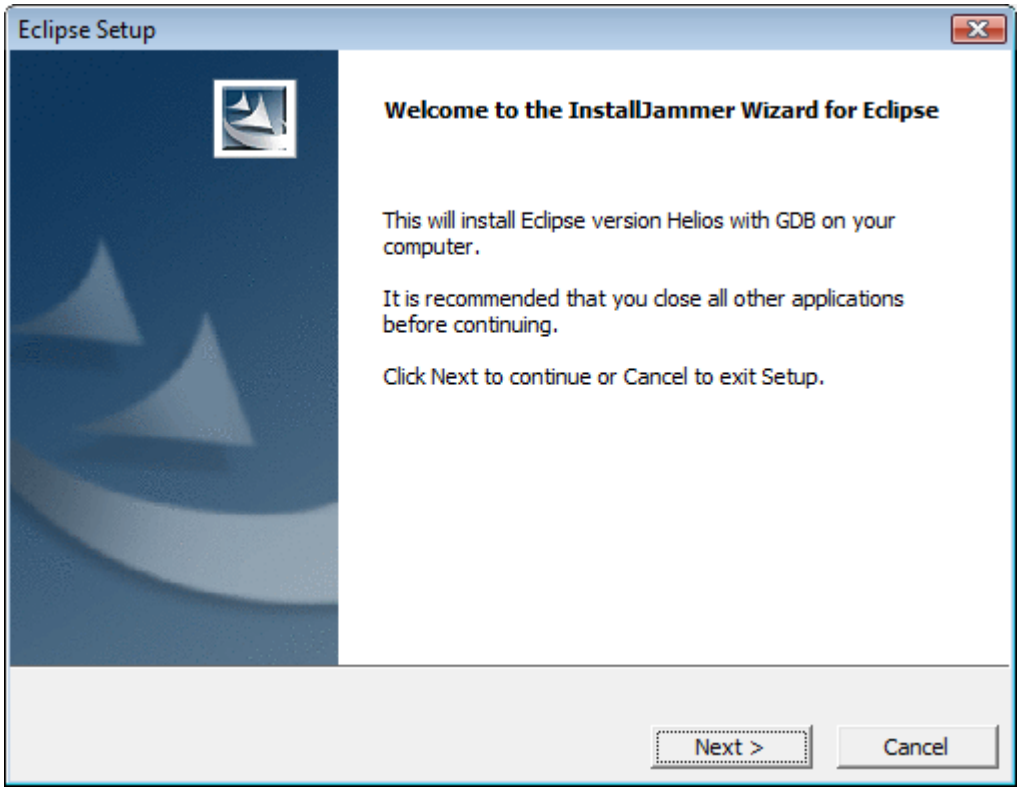


Figure 31: Install ICP SDK step 18

Select ‘Next >’ to continue.

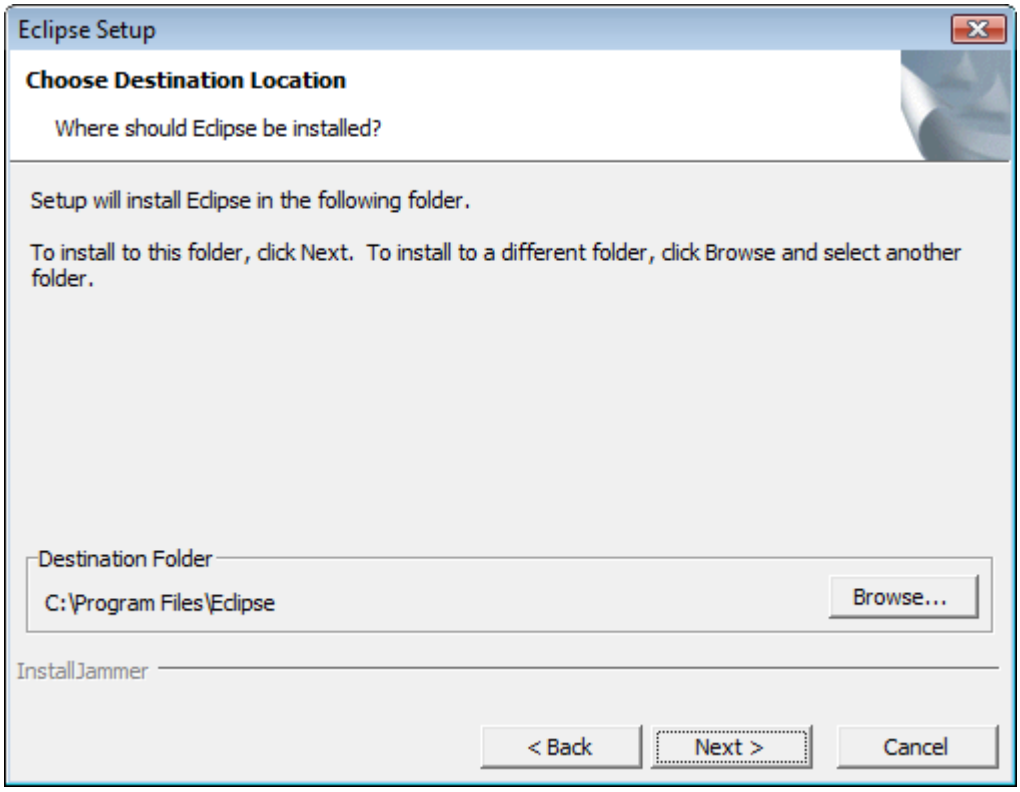


Figure 32: Install ICP SDK step 19

Select ‘Next >’ to continue.

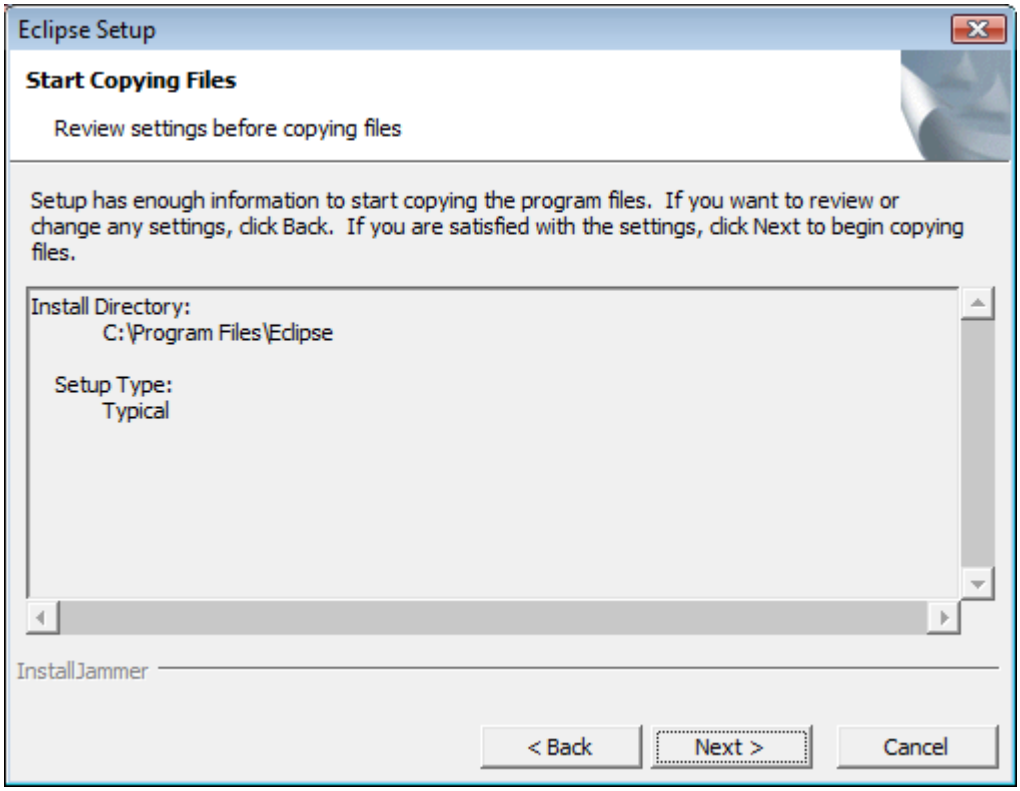


Figure 33: Install ICP SDK step 20

Select ‘Next >’ to continue.

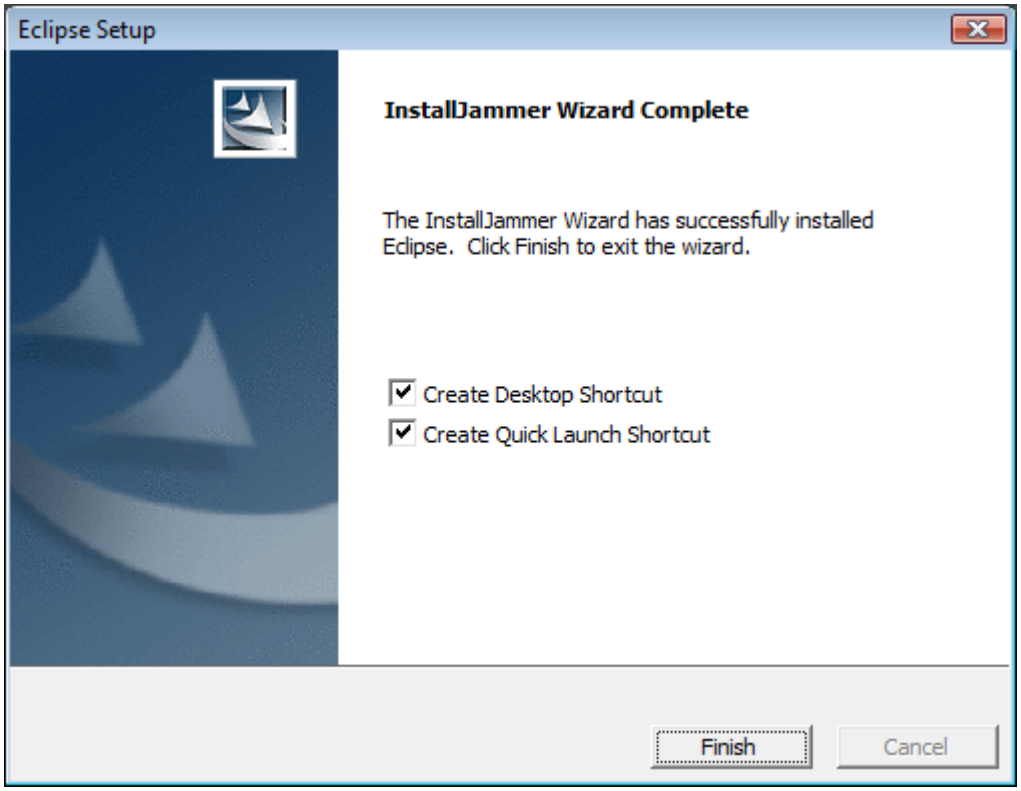


Figure 34: Install ICP SDK step 21

Select ‘Next >’ to continue.

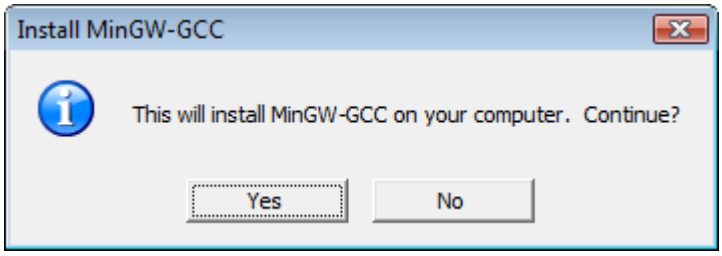


Figure 35: Install ICP SDK step 22

Now, MinGW is being installed.

Select ‘Yes’ to continue.

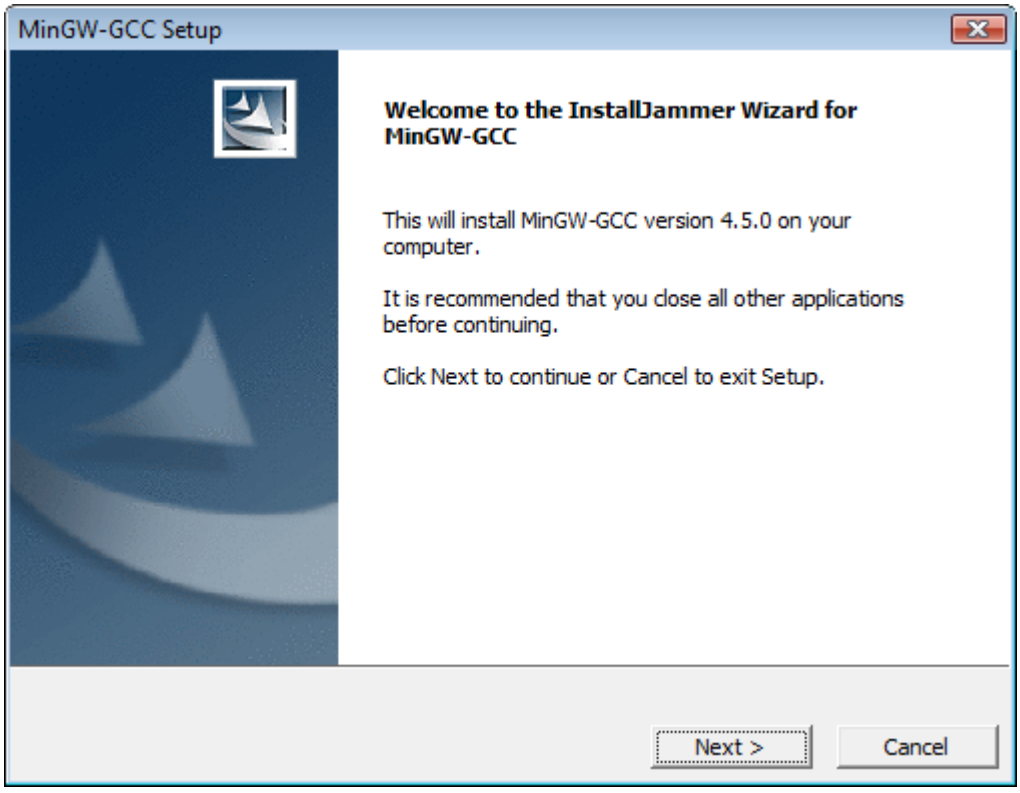


Figure 36: Install ICP SDK step 23

Select ‘Next >’ to continue.

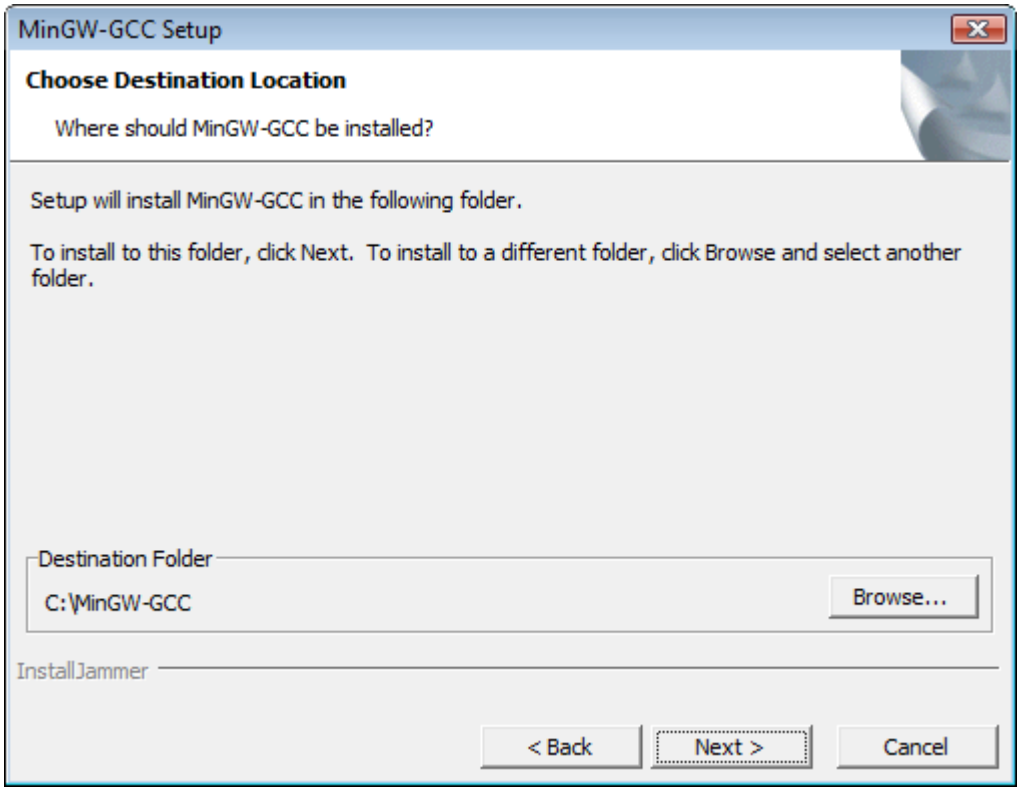


Figure 37: Install ICP SDK step 24

Select ‘Next >’ to continue.

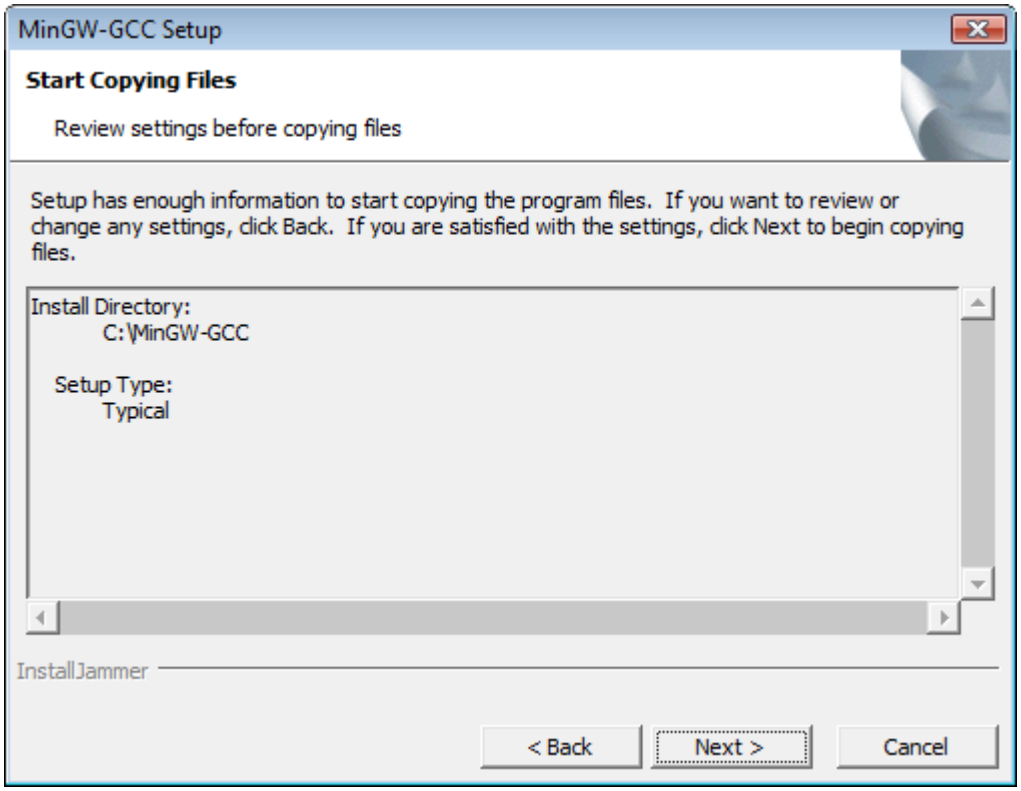


Figure 38: Install ICP SDK step 25

Select ‘Next >’ to continue.

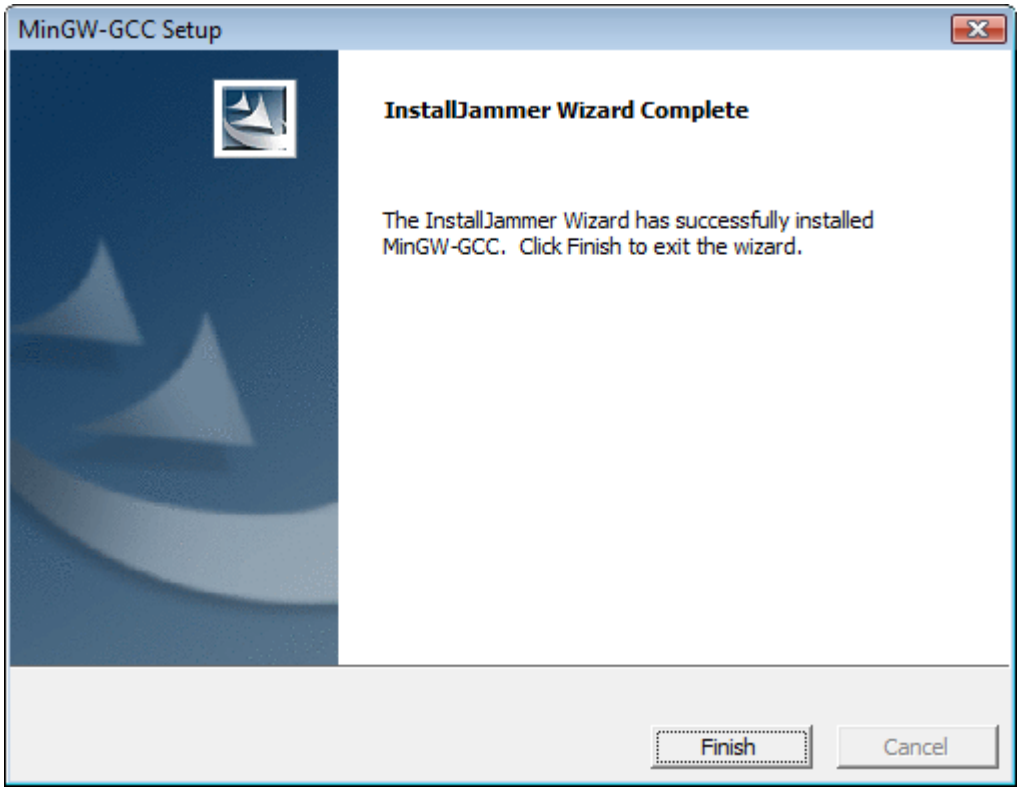


Figure 39: Install ICP SDK step 26

MinGW is now installed.

Select ‘Finish’ to continue.

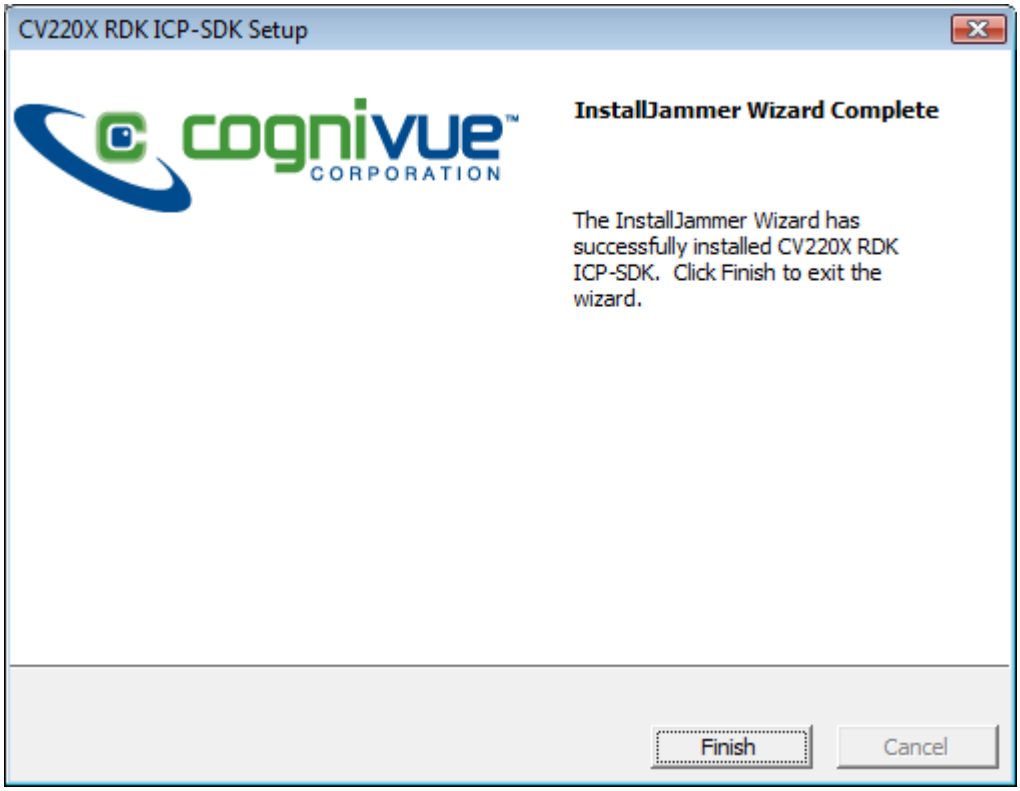


Figure 40: Install ICP SDK step 27

The package content is now installed. It is now possible to create a first application and compile it for the development kit.

Select ‘Finish’ to continue.

4 Programming on the Development Platforms

Creating applications running on the development platform is quite easy using Image Cognition Processor Software Development Kit (ICPSDK) and provided tools.

The easiest way to generate a binary file of the code developed for your target development platform is to use an existing project and adapt to your needs.

Steps to generate a binary file:

- Setup project
- Update makefile
- Optional hardwareinit.h
- Compilation/linkage

4.1 Set up project

It is suggested to place your code under 'ICPSDK_location' \Applications\'companyname'\'applicationname'

like the sample code provided: 'C:\CV220X_RDK_ICP_SDK\Applications\Cognivue\SobelApp'

where:

- 'C:\CV220X_RDK_ICP_SDK' is the ICPSDK path
- 'Cognivue' is the company name
- 'SobelApp' is the application name

At this location, it is suggested to place the following files:

- Makefile : file containing the instruction for compilation
- 'app'.cmm : file for Lauterbach Trace32 debugger (optional if using this debugger)

And the following sub-folders:

- inc : where the application header files (.h/.hpp) are located
- src : where the application code files (.c/.cpp) are located, may contain a sub-folder SIMD with APEX kernel source files (.sel)

4.2 Update makefile

The file 'Makefile' is a text file containing the description of the application for compilation. It can be opened and modified by a text editor.

Here are the main sections to update for a specific application:

```
# SDKROOT, the location of ICPSDK on the PC
SDKROOT  = C:/CV220X_RDK_ICP_SDK
# APPROOT, the relative location of the application root
APPROOT  = .
#APPNAME, the name of the application, the binary will have this name with '.bin' extension
# here the binary file will be: 'AppName.bin'
APPNAME  = AppName
.
.
# CPPSRCS += , add a source code file to be added to the application (located in 'src' sub-folder)
CPPSRCS  += Main.cpp
CPPSRCS  += Task1.cpp
CPPSRCS  += Task2.cpp
.
.
#SELS += , add a APEX kernel source code file to be added (located in 'src/SIMD' sub-folder)
SELS += MNP_1D_FLTV.sel
```

4.3 Optional rdk_hw_init.cpp

ICP SDK provides a default configuration to manage the peripherals of CV220x chip on the development plateform. The default configuration is provided in the following files:

- C:\CV220X_RDK_ICP_SDK\Drivers\Common\src\CallbackInitializeHardwareRDK.cpp , this is for the RDK
- C:\CV220X_RDK_ICP_SDK\Drivers\Common\src\CallbackInitializeHardwareEVB.cpp , this is for the PDK

It is possible to customize the configuration by providing an alternate description in a file in the current application source folder ' src/ rdk_hw_init.cpp'

It is necessary to add it to the Makefile so it supersedes the default configuration.

```
CPPSRCS  += rdk_hw_init.cpp
```

4.4 Compilation/linkage

Compilation and linkage are achieve by invoking 'cs-make' in the root folder of the application.

‘cs-make’ will use the information in the file ‘Makefile’ to generate the binary for the application.

It might be necessary to specify the type of CV220x chip used on the platform. By default, ICPSDK compiles for the RDK, so for the PDK or hardware using a CV2201, the following command will set the chip:

‘cs-make CHIP=2201’

It is possible to request a clean up of the intermediate file by adding ‘clean’, in this case, cs-make will not build unless the command ‘build’ is added as well:

‘cs-make clean build CHIP=2201’

4.5 Compilation options

To modify the compiler options such as the optimization level, it can be done with the following file:

‘C:\CV220X_RDK_ICP_SDK\ Makefile.gnu’ , this file is compiler specific, this one dedicated to the GNU tool chain (provided with ICPSDK), if you are using a different compiler, you need to create the appropriate equivalent of this file.

The variable OFLAGS sets the optimization level, default is 2. To debug it might be convenient to use an optimization level of 0 that would compile the code exactly as written with no optimizations.

OFLAGS += -O2

It is possible to further change options, consult compiler and debugger manuals for further information.

5 Running Applications on the Development Platforms

It is possible to run a custom program on the development platform, either from JTAG or from non-volatile memory.

5.1 Reference Development Kit Programming (RDK)

The RDK is using a multi-boot program offering the selection of which application, stored on the SD card, to run at boot up.

The multi-boot program is looking for a file config.txt to learn which application to run. The selection is done through the buttons and joystick on the RDK.

Function	Sample content of config.txt
Required	0x00 0x0A
Default application, no button pressed	C:\DefaultApp.bin
Keyout0 pressed	C:\App1.bin
Keyout1 pressed	C:\App2.bin
Keyout2 pressed	C:\App3.bin
Keyout3 pressed	C:\App4.bin
Joystick Down pressed	C:\App5.bin
Joystick Left pressed	C:\App6.bin
Joystick UP pressed	C:\App7.bin
Joystick Right pressed	C:\App8.bin
Joystick pressed	C:\App9.bin

The SD card content can be read from a PC with an appropriate hardware reader. The SD card should be formatted in FAT32, capacity should be up to 32GB.

Config.txt can be edited by most text editor such as notepad on MS Windows.

The binary files *.bin can be copied to the SD card from the development PC that generated them.

Cycle the RDK power to initiate the boot-up sequence and press the desired button to start application.

5.2 Automotive and Consumer PDK

If you need this functionality, please contact factory.

6 JTAG Debugging

It is recommended to use a JTAG debugger for development. The JTAG debugger, not provided by CogniVue, needs to be compatible for ARM926™ processor. The probe will be connected to the development platform via the JTAG connector directly or through an adaptor board (provided with development platform).

There are quite a few vendors and models available on the market. CogniVue has had good success with Lauterbach Trace32® and will present instructions for this model only.

NOTE: make sure the development platform is set to boot with JTAG.

- For RDK, see: 2.1.4.1 Power-up Configuration settings (JTAG)
- For Automotive PDK, see: 2.2.3.1 Switch configuration
- For Consumer PDK, see: 2.3.3.1 Switch configuration

6.1 JTAG connector

A standard 2x10 connector is available for ARM debugger connectivity. The debugger connectivity allows any third party debugger tools to connect to the CV220x ARM926 for software development.

Table 18: JTAG debugger pinout

Name	Pin	Pin	Name
3.0V	1	2	3.0V
nTRST	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
RTCK	11	12	GND
TDO	13	14	GND
RESETn	15	16	GND
	17	18	GND
	19	20	GND

6.2 Lauterbach Trace32

Sample code is provided with command files for Lauterbach Trace32 debugger; these files have the extension .cmm.

The following are the steps to run an application with the debugger:

- Connect the debugger on the JTAG connector, power the probe, and then the development platform.

- Start the Trace32 debugger on your PC.
- From the file menu, select Edit Batchfile.
- Browse to your project folder, select '*filename*'.cmm (If not present, copy a .cmm file from another location and copy it in your project; change the binary name to your application name.)
- Click the Save+Do button; the application will load.
- Click Go at the source window; the application will run.

It might be necessary to adapt the cmm file to your setup. This is done by editing the cmm file, with a text editor or with the debugger directly, through the following lines at the beginning of the file:

; SDKROOT, location of ICPSDK on PC

&SDKROOT="C:/CV220X_RDK_ICP_SDK"

; BINARY_NAME, application name 'app'.bin, will be expected to be found in 'bin' sub-folder to where the cmm is located.

&BINARY_NAME="CaptureApp"

; Chip = 2201 / 2202 / 2207, for RDK 2202, PDK 2201

&Chip="2202"

; Load = rvds40 / gnuarm

&Load="gnuarm"

Refer to Lauterbach Trace32 manuals for further reference.

7 Revision History

Revision	Details of Change	Date
1	Initial release	11/2013