AN3252

How to Use the SAMA5D2 I2C Under Linux®

Introduction

This application note describes how to get started using the SAMA5D2 I2C under Linux.

In most cases, I2C devices are controlled by a kernel driver. However, it is also possible to access all devices on an adapter from user space through the I2C dev interface. I2C dev is a character device node file that can be accessed by read(), write() and ioctl(). The interface of each I2C bus can be exported to user space through its own I2C dev device node.

I2C dev application demo code is provided in the section Application.

Reference Documents

Title	Reference	Available
SAMA5D2 Series Datasheet	DS60001476	https://www.microchip.com/design-centers/32-bit-mpus
SAMA5D27 SOM1 Kit1 User Guide	DS50002667	https://www.microchip.com/DevelopmentTools/ProductDetails/PartNO/ATSAMA5D27-SOM1-EK1

Prerequisites

- Hardware
  - PC
  - SAMA5D27 SOM1 Evaluation Kit (Part Number: ATSAMA5D27-SOM1-EK1)
  - SDCard
  - mikroBUS™ Thermo 5 Click board™
- Software
  This demo runs on the AT91 Linux platform built by Buildroot. The first step is to set up the AT91 Buildroot development environment. Refer to the web site: http://www.at91.com/linux4sam/bin/view/Linux4SAM/BuildRoot

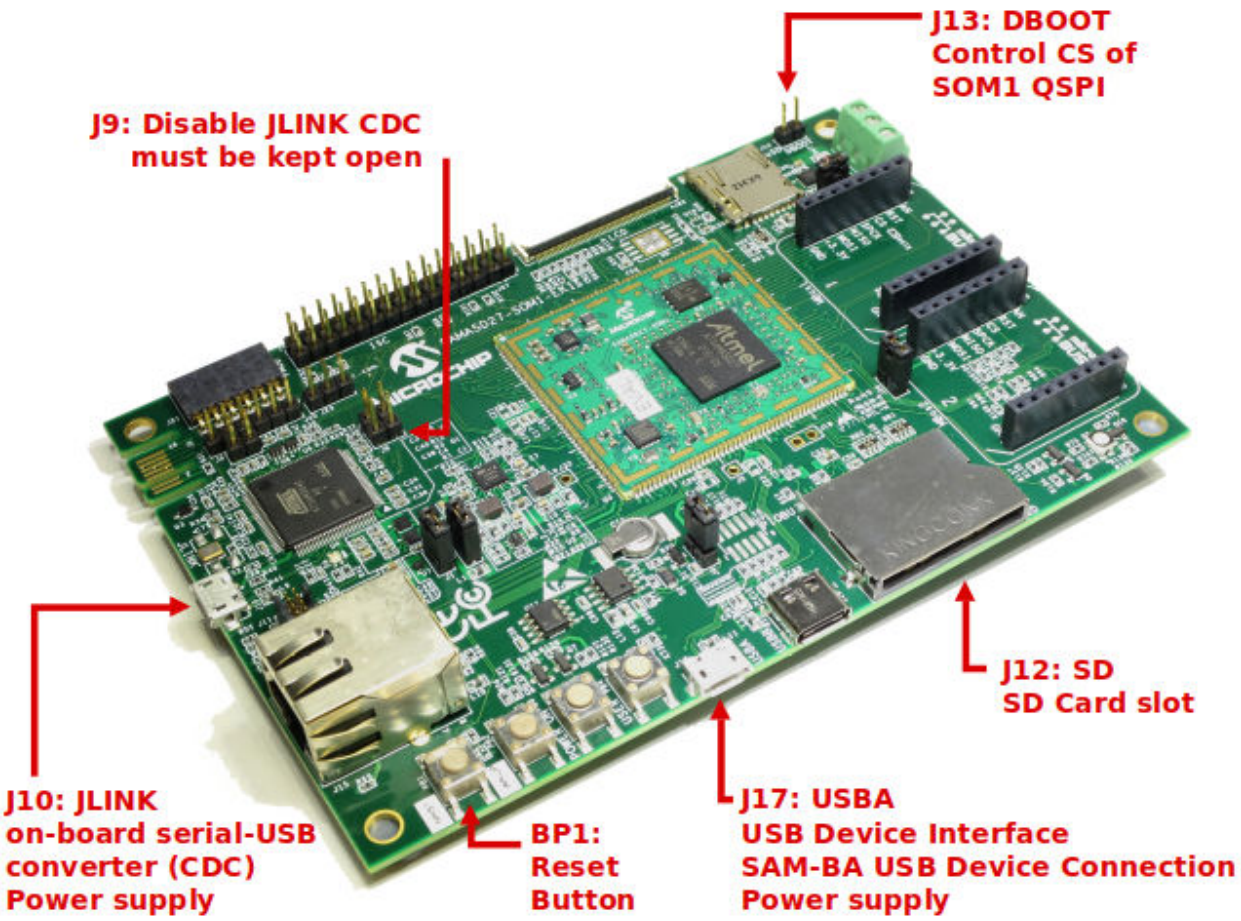
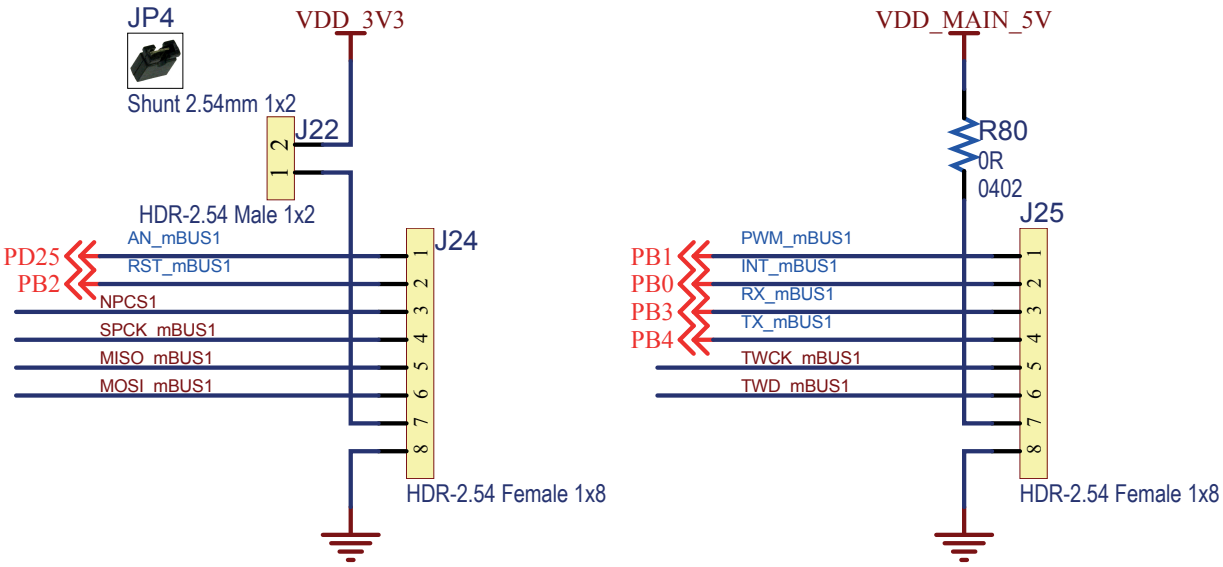

J13: DBOOT
Control CS of
SOM1 QSPI
J9: Disable JLINK CDC
must be kept open
J12: SD
SD Card slot
J10: JLINK
on-board serial-USB
converter (CDC)
Power supply
BP1:
Reset
Button
J17: USBA
USB Device Interface
SAM-BA USB Device Connection
Power supply

Table of Contents

1. Hardware Design

1.1 Interface



The mikroBUS1 connector is used for easy testing and monitoring.

To control the mikroBUS1 I2C on Linux, FLEXCOM1 (I2C mode) is connected to the I2C bus of the mikroBUS1 interface on the SAMA5D27-SOM1-EK as described below:

FLEXCOM1 I2C

- FLEXCOM1_IO1 → PA23 → TWCK_mBUS1
- FLEXCOM1_IO0 → PA24 → TWD_mBUS1

For more details about the pin multiplexing of the SAMA5D2, refer to the table “Pin Description (all packages)” in the SAMA5D2 data sheet.

FLEXCOM I/O Lines Description

Name	Description			Type
	USART/UART	SPI	TWI	
FLEXCOM_IO0	TXD	MOSI	TWD	I/O
FLEXCOM_IO1	RXD	MISO	TWCK	I/O
FLEXCOM_IO2	SCK	SPCK	–	I/O
FLEXCOM_IO3	CTS	NPCS0/NSS	–	I/O
FLEXCOM_IO4	RTS	NPCS1	–	O

1.2 Device Connection

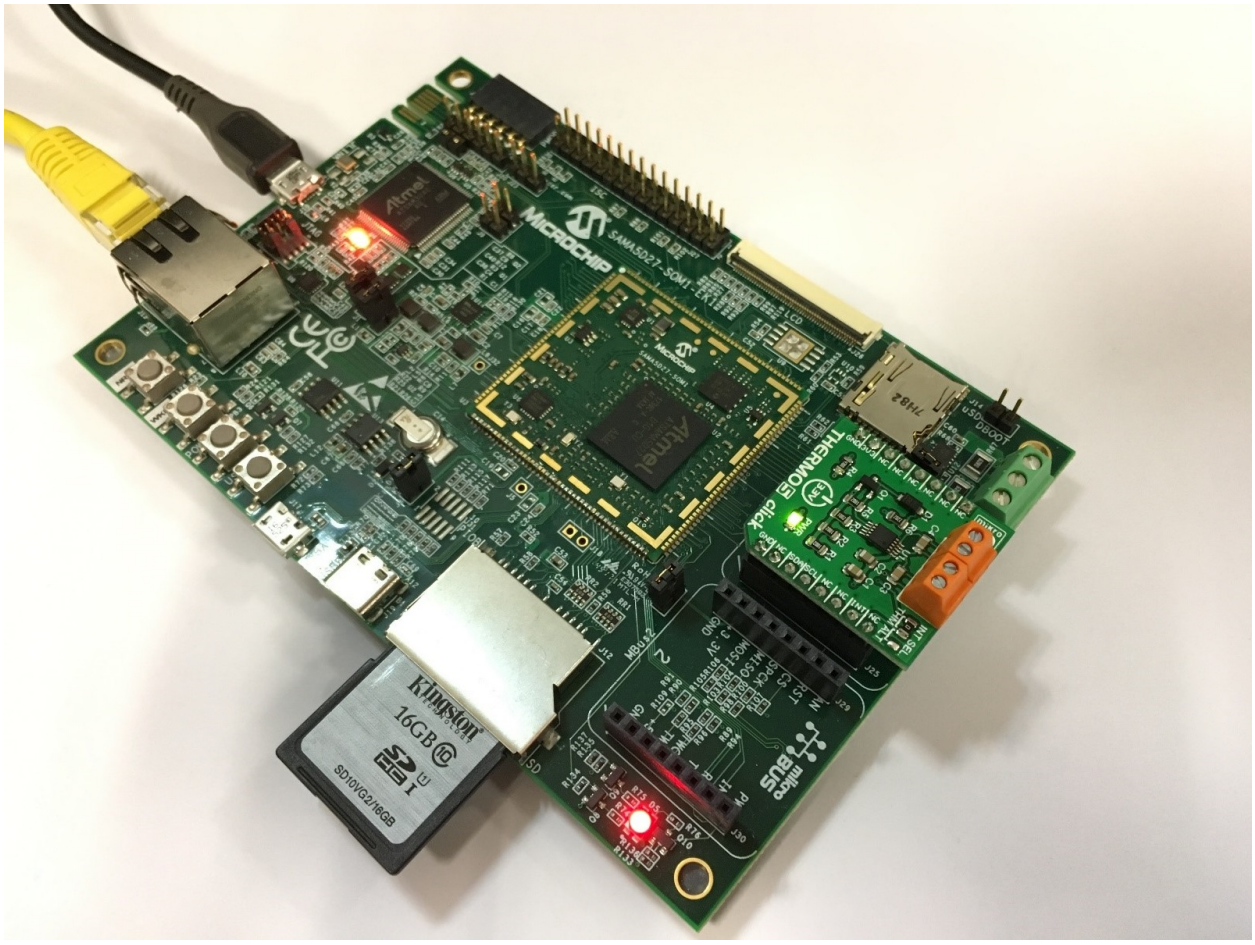
A Thermo 5 click board is used in this demo. It must be connected to the SAMA5D27-SOM1-EK mikroBUS1.

The Thermo 5 click board measures temperature in the default range of 0°C to 127°C and in the extended range of -64°C to 191°C with ±1°C accuracy. It features the **EMC1414** temperature sensor.

Refer to https://www.mikroe.com/thermo-5-click.

Figure 1-1. Thermo 5 Click Board

Figure 1-2. Thermo 5 On the SAMA5D27-SOM1-EK1 Evaluation Board



2. Software Design

The Microchip Linux platform was built using Buildroot with the following configuration:

atmel_sama5d27_som1_ek_mmc_dev_defconfig

The I2C bus driver works under this default configuration.

There are two methods to access the I2C bus driver:

- In kernel space:
  - Register your own I2C driver via i2c_add_driver() interface, then access the I2C bus driver via struct i2c_client handle.
- In user space:
  - Enable I2C_CHARDEV kernel feature, then access the I2C bus driver via /dev/i2c-X device node.

I2C_CHARDEV is a good choice because the full code runs in user space (easy development) with the default configuration I2C_CHARDEV feature enabled.

2.1 Device Tree

- Action: no need to change
- Location: buildroot-at91/output/build/linux-linux4sam_6.0/arch/arm/boot/dts
- Sources:
  - sama5d2.dtsi
  - at91-sama5d27_som1_ek.dts

Device tree for FLEXCOM1 in sama5d2.dtsi:

```
flx1: flexcom@f8038000 {
    compatible = "atmel,sama5d2-flexcom"; // specify which driver will be used for this
                                          // FLEXCOM device
    reg = <0xf8038000 0x200>; // FLEXCOM1 base address is 0xf8038000, size is 0x200

    clocks = <&flx1_clk>; // definition for FLEXCOM1 clock source
    #address-cells = <1>;
    #size-cells = <1>;
    ranges = <0x0 0xf8038000 0x800>;
    status = "disabled"; // default disabled, and will be replaced with “okay”
};

flx1_clk: flx1_clk {
    #clock-cells = <0>;
    reg = <20>; // PID for FLEXCOM1 is 20, this definition of offset will be used to enable
                // FLEXCOM1 clock in PMC
    atmel,clk-output-range = <0 83000000>; // FLEXCOM1 input clock, max frequency is 83MHz
};
```

Device tree for I2C device function in at91-sama5d27_som1_ek.dts:

```
aliases {
    serial0 = &uart1; /* DBGU */
    serial1 = &uart4; /* mikro BUS 1 */
    serial2 = &uart2; /* mikro BUS 2 */
    i2c1  = &i2c1;
    i2c2  = &i2c2; // the aliases of FLEXCOM1 I2C is i2c2, then FLEXCOM1 I2C will be
           // registered as I2C adapter 2
};

flx1: flexcom@f8038000 {
    atmel,flexcom-mode = <ATMEL_FLEXCOM_MODE_TWI>; // specify I2C mode for this FLEXCOM port
    status = "okay"; // enable this device

    i2c2: i2c@600 {
        compatible = "atmel,sama5d2-i2c"; // specify which driver will be used for this I2C
                                          // device
        reg = <0x600 0x200>; // register offset address for I2C in FLEXCOM1 is 0x600,
                        //  size is  0x200
        interrupts = <20 IRQ_TYPE_LEVEL_HIGH 7>; // PID for FLEXCOM4 is 23, high level
                                    // triggered, priority is 7
                                    // used to configure FLEXCOM4 interrupt in AIC
        dmas = <0>, <0>; // DMA feature wasn't enabled
        dma-names = "tx", "rx";
        #address-cells = <1>;
        #size-cells = <0>;
        clocks = <&flx1_clk>; // definition for FLEXCOM1 clock source
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_mikrobus_i2c>; // pin definition for I2C
        atmel,fifo-size = <16>; // specify size of fifo is 16
        status = "okay"; // enable this I2C device
    };
};

pinctrl_mikrobus_i2c: mikrobus1_i2c {
    pinmux = <PIN_PA24__FLEXCOM1_IO0>, // the mux of PA24 will be switched to FLEXCOM1_IO0
        <PIN_PA23__FLEXCOM1_IO1>; // the mux of PA23 will be switched to FLEXCOM1_IO1
    bias-disable; // pull up/down feature disabled
};
```

2.2 Kernel

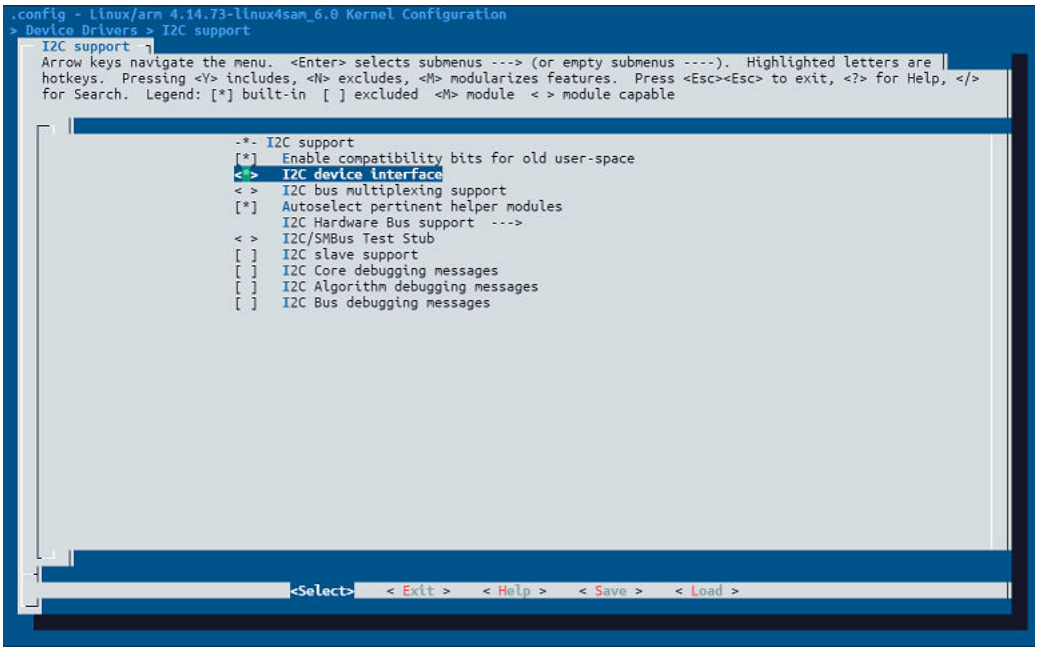
- Action: no need to change
- Location: buildroot-at91/output/build/linux-linux4sam_6.0/
- Defconfig: sama5_defconfig
- Driver files:
  - drivers/i2c/i2c-core-base.c
  - drivers/i2c/i2c-dev.c
  - drivers/i2c/busses/i2c-at91.c

Check the kernel configuration for the ADC function:

```
user@at91:~/buildroot-at91$ make linux-menuconfig
```

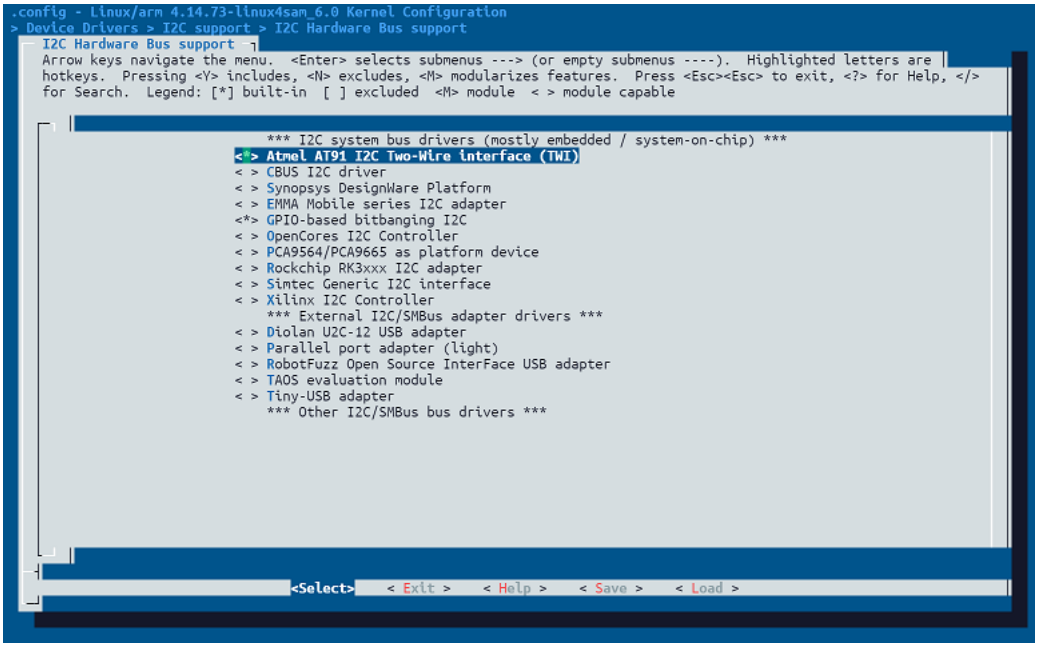
Device Drivers > I2C support > I2C device interface
Driver for I2C_CHARDEV device.

In the default configuration, this item should be selected.



Device Drivers > I2C support > Atmel AT91 I2C Two-Wire interface (TWI)
Driver for AT91 I2C controller.

In the default configuration, this item should be selected.



2.3 Rootfs

- Action: no need to change
- Location: buildroot-at91/output/images/rootfs.tar

As discussed in the section Device Tree, the FLEXCOM1 I2C is registered as an I2C adapter 2.

The following device node is used to access the FLEXCOM1 I2C bus driver:

- /dev/i2c-2

2.4 Application

This section provides a C language demo i2c_dev.c to read temperature data from the Thermo 5 click board via /dev/i2c-2 node

How to Compile

```
user@at91:~$ buildroot-at91/output/host/bin/arm-buildroot-linux-uclibcgnueabihf-gcc
i2c_dev.c -o i2c_test
```

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.h>

#define DEV_I2C "/dev/i2c-2"

#define SLAVE_ADDR 0x4C /* EMC1414 I2C slave address */
//#define COMBINED_TRANSCTION

int main(int argc, char *argv[])
{
    int fd;
    int ret;
    unsigned char buf[2];

    // open device node
    fd = open(DEV_I2C, O_RDWR);
    if (fd < 0) {
        printf("ERROR open %s ret=%d\n", DEV_I2C, fd);
        return -1;
    }

    if (ioctl(fd, I2C_SLAVE, SLAVE_ADDR) < 0) {
        printf("ERROR ioctl() set slave address\n");
        return -1;
    }

#ifdef COMBINED_TRANSCTION
    struct i2c_rdwr_ioctl_data data;
    struct i2c_msg messages[2];

    // Set conversion rate
    buf[0] = 0x04; // Conversion rate register address
    buf[1] = 0x04; // Set conversion rate to 1 second
    messages[0].addr  = SLAVE_ADDR; //device address
    messages[0].flags = 0; //write
    messages[0].len   = 2;
    messages[0].buf   = buf; //data address

    data.msgs  = &messages[0];
    data.nmsgs = 1;
    if (ioctl(fd, I2C_RDWR, &data) < 0) {
        printf("ERROR ioctl() conversion rate\n");
        return -1;
    }

    // Read temperature
    buf[0] = 0x00; // Internal Diode High Byte register address
    buf[1] = 0;    // clear receive buffer
    messages[0].addr  = SLAVE_ADDR; //device address
    messages[0].flags = 0; //write
    messages[0].len   = 1;
    messages[0].buf   = &buf[0]; //data address
```

```
    messages[1].addr  = SLAVE_ADDR; //device address
    messages[1].flags = I2C_M_RD; //read
    messages[1].len   = 1;
    messages[1].buf   = &buf[1];

    data.msgs  = messages;
    data.nmsgs = 2;
    while (1) {
        if (ioctl(fd, I2C_RDWR, &data) < 0) {
            printf("ERROR ioctl() read data\n");
            return -1;
        }

        printf("Temperature is %d\n", buf[1]);
        sleep(1);
    }

#else
    // Set conversion rate
    buf[0] = 0x04; // Conversion rate register address
    buf[1] = 0x04; // Set conversion rate to 1 second
    ret = write(fd, buf, 2);
    if (ret != 2) {
        printf("ERROR write() conversion rate\n");
        return -1;
    }

    // Read temperature
    // Set internal address register pointer
    buf[0] = 0x00; // Internal Diode High Byte register address
    ret = write(fd, &buf[0], 1);
    if (ret != 1) {
        printf("ERROR write() register address\n");
        return -1;
    }

    while (1) {
        // Read temperature
        // Read data
        buf[1] = 0; // clear receive buffer
        ret = read(fd, &buf[1], 1);
        if (ret != 1) {
            printf("ERROR read() data\n");
            return -1;
        }

        printf("Temperature is %d\n", buf[1]);
        sleep(1);
    }
#endif

    // close device node
    close(fd);

    return 0;
}
```

3. Hands-On

Copy i2c_test app to the target and execute it. The temperature data is then printed out.

```
# chmod +x i2c_test
# ./i2c_test
Temperature is 26
Temperature is 26
Temperature is 26
Temperature is 26
Temperature is 26
Temperature is 26
```

4. Tools and Utilities

I2c-tools is a tool for I2C bus testing which is included in Buildroot.

In the default Buildroot configuration, this tool is selected.

```
user@at91:~/buildroot-at91$ make menuconfig
```

Target packages > Hardware handling > i2c-tools
There are several commands in `i2c-tools`:

```
# i2cdetect
Error: No i2c-bus specified!
Usage: i2cdetect [-y] [-a] [-q|-r] I2CBUS [FIRST LAST]
       i2cdetect -F I2CBUS
       i2cdetect -l
  I2CBUS is an integer or an I2C bus name
  If provided, FIRST and LAST limit the probing range.
# i2cdetect -y 3
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: UU 51 52 53 54 55 56 57 -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --

# i2cdump
Error: No i2c-bus specified!
Usage: i2cdump [-f] [-y] [-r first-last] I2CBUS ADDRESS [MODE [BANK [BANKREG]]]
  I2CBUS is an integer or an I2C bus name
  ADDRESS is an integer (0x03 - 0x77)
  MODE is one of:
    b (byte, default)
    w (word)
    W (word on even register addresses)
    s (SMBus block)
    i (I2C block)
    c (consecutive byte)
    Append p for SMBus PEC
# i2cdump -f -y 3 0x50
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
10: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
20: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
30: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
40: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
50: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
60: 20 4d 43 48 49 50 20 52 46 4f 12 02 42 42 31 ff     MCHIP RFO??BB1.
70: 00 00 44 32 37 2d 53 4f 4d 31 2d 45 4b 31 44 f7    ..D27-SOM1-EK1D?
80: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
90: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
a0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
b0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
c0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
d0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
e0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    ................
f0: ff ff ff ff ff ff ff ff ff ff 54 10 ec 33 c5 7d    ..........T??3?}

# i2cget
Usage: i2cget [-f] [-y] I2CBUS CHIP-ADDRESS [DATA-ADDRESS [MODE]]
  I2CBUS is an integer or an I2C bus name
  ADDRESS is an integer (0x03 - 0x77)
  MODE is one of:
    b (read byte data, default)
    w (read word data)
    c (write byte/read byte)
Append p for SMBus PEC
# i2cget -f -y 3 0x50 0x60
0x20
```

```
# i2cget -f -y 3 0x50 0x61
0x4d

# i2cset
Usage: i2cset [-f] [-y] [-m MASK] [-r] I2CBUS CHIP-ADDRESS DATA-ADDRESS [VALUE] ... [MODE]
  I2CBUS is an integer or an I2C bus name
  ADDRESS is an integer (0x03 - 0x77)
  MODE is one of:
    c (byte, no value)
    b (byte data, default)
    w (word data)
    i (I2C block data)
    s (SMBus block data)
Append p for SMBus PEC
WARRING: DO NOT modify any data in Eeprom of SAMA5D27_SOM1

# i2ctransfer
Usage: i2ctransfer [-f] [-y] [-v] [-V] I2CBUS DESC [DATA] [DESC [DATA]]...
  I2CBUS is an integer or an I2C bus name
  DESC describes the transfer in the form: {r|w}LENGTH[@address]
    1) read/write-flag 2) LENGTH (range 0-65535) 3) I2C address (use last one if omitted)
  DATA are LENGTH bytes for a write message. They can be shortened by a suffix:
    = (keep value constant until LENGTH)
    + (increase value by 1 until LENGTH)
    - (decrease value by 1 until LENGTH)
    p (use pseudo random generator until LENGTH with value as seed)
```

Example (bus 0, read 8 byte at offset 0x64 from EEPROM at 0x50): `# i2ctransfer 0 w1@0x50 0x64 r8`

Example (same EEPROM, at offset 0x42 write 0xff 0xfe ... 0xf0): `# i2ctransfer 0 w17@0x50 0x42 0xff-`

5. Microchip Peripheral I/O Python® (MPIO)

The Microchip Peripheral I/O (MPIO) Python package provides easy access to various hardware peripherals found on Microchip MPU processors and evaluation boards running Linux. The API is clean, consistent, flexible, documented, and well tested. It makes navigating and exercising even the most complex hardware peripherals a trivial task.

For more information, see https://github.com/linux4sam/mpio. Code examples showing how to work with the MPIO interface modules are provided in the folder mpio/examples.

5.1 MPIO in buildroot

In order to benefit from MPIO in your buildroot configuration, follow the steps below:

1. Enable Python

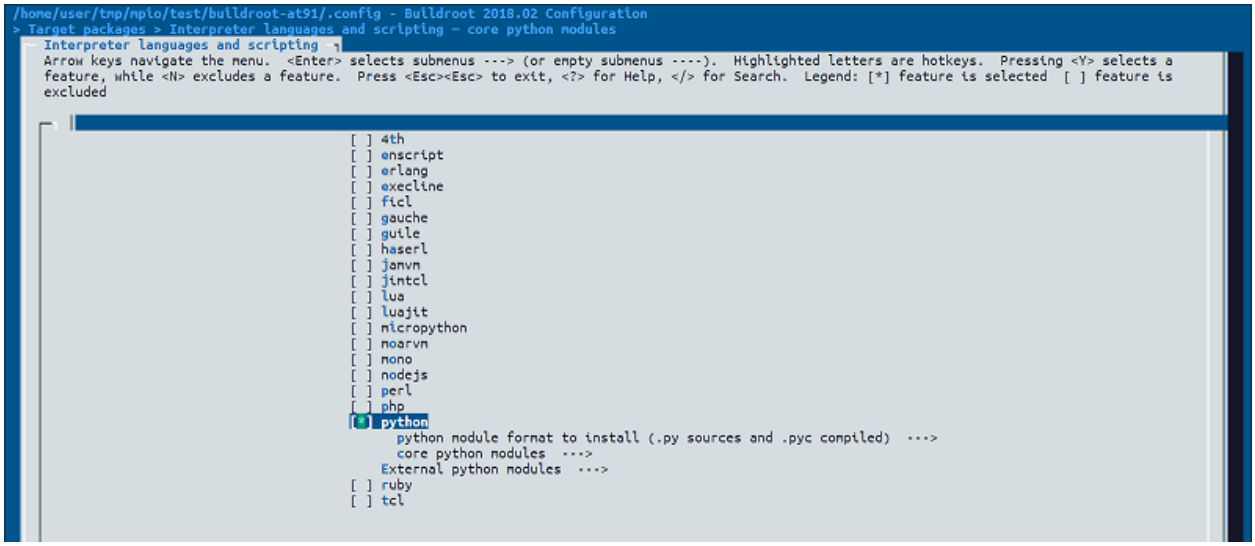
```
user@at91:~/buildroot-at91$ make menuconfig
```

Select “**python**” to enable python support:

- Target packages > Interpreter languages and scripting > [*] **python**

Then enter “**python module format to install**” and select “**.py sources and .pyc compiled**”.

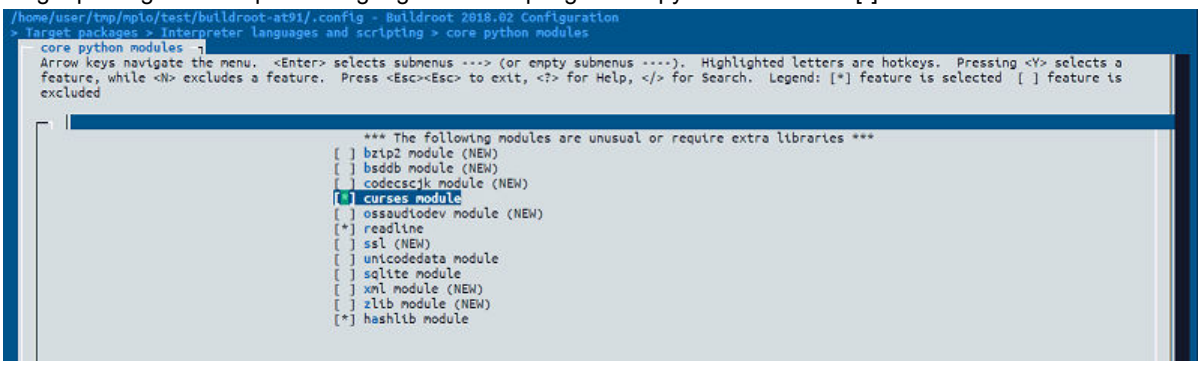
- Target packages > Interpreter languages and scripting > python > python module format to install > **.py sources and .pyc compiled**



Some additional python modules must be selected. Enter “**core python modules**” and select “**curses module**”, “**readline**” and “**hashlib module**”.

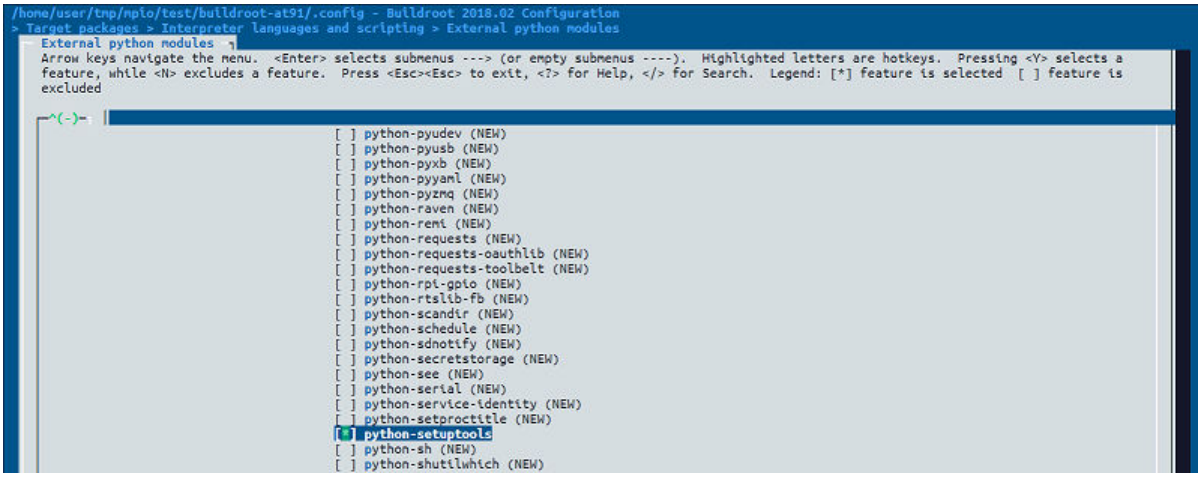
- Target packages > Interpreter languages and scripting > core python modules > [*] **curses module**
- Target packages > Interpreter languages and scripting > core python modules > [*] **readline**

- Target packages > Interpreter languages and scripting > core python modules > [*] **hashlib module**



Enter “**External python modules**” and select “**python-setuptools**”.

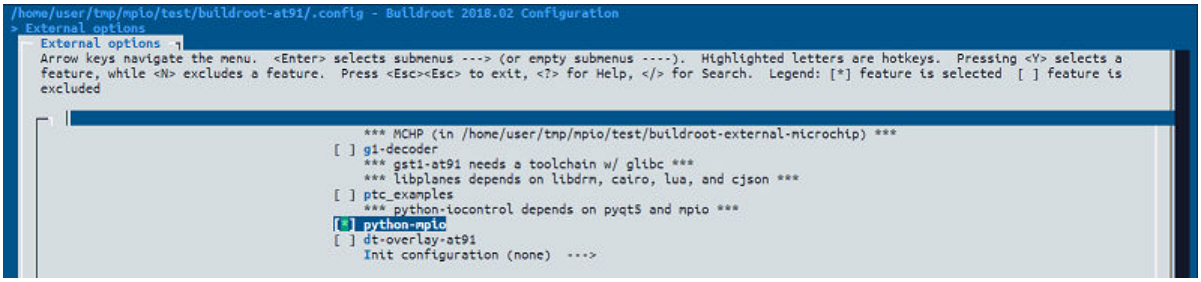
- Target packages > Interpreter languages and scripting > External python modules > **[*] python-setuptools**



2. Enable the MPIO Module

Enter “**External options**” and select “**python-mpio**”.

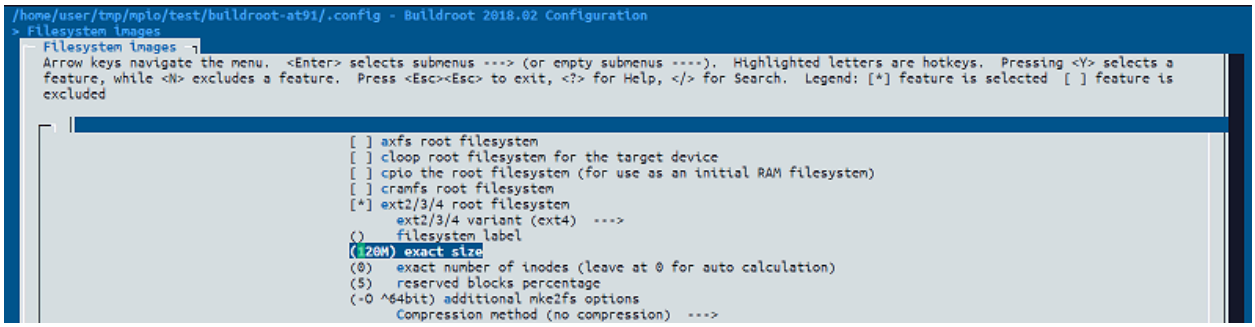
- External options > **[*] python-mpio**



3. Finish the buildroot Configuration and Build

Enter “**Filesystem images**” and set the exact size of rootfs to 120MB.

- Filesystem images > **(120M) exact size**



After saving, the following new settings are added to the configuration file of buildroot:

……

BR2_PACKAGE_PYTHON=y

BR2_PACKAGE_PYTHON_PY_PYC=y

BR2_PACKAGE_PYTHON_CURSES=y

BR2_PACKAGE_PYTHON_READLINE=y

BR2_PACKAGE_PYTHON_HASHLIB=y

BR2_PACKAGE_PYTHON_SETUPTOOLS=y

BR2_PACKAGE_PYTHON_MPIO=y

BR2_TARGET_ROOTFS_EXT2_SIZE="120M"

……

Then re-configure and build buildroot:

user@at91:~/buildroot-at91$ **make atmel_sama5d27_som1_ek_mmc_dev_defconfig**

user@at91:~/buildroot-at91$ **make**

5.2 Examples

After building successfully, burn your SD card with buildroot-at91/output/images/**sdcard.img**.

Execute the python codes on the target board, for example:

./adc2.py DEVICE

./gpio1.py PIN

./pwm_led.py DEVICE CHANNEL

……

Note: The python example code can be found in https://github.com/linux4sam/mpio/examples

6. Revision History

6.1 Rev. A - 09/2019

First issue.

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with

your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-5046-7

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: http://www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820