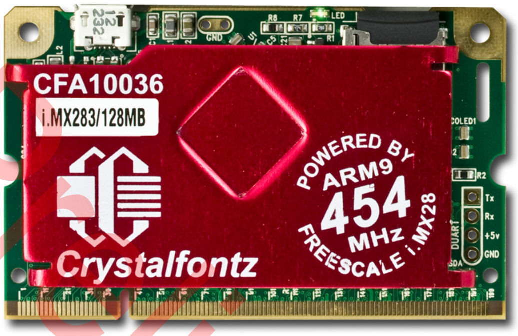
Crystalfontz America, Incorporated

CFA10036 ARM9 + Linux SOM Family Guide



CFA10036 shown with i.MX283 processor

DISCLAIMER
The information in this Guide is preliminary, incomplete, and has not yet been fully validated.

CFA10036 ARM9 + Linux SOM Family Guide Preliminary Release 2013-11-20
for
CFA10036 Hardware Version: v1.0

Crystalfontz America, Incorporated

12412 East Saltese Avenue
Spokane Valley, WA 99216-0357

Phone: 888-206-9720
Fax: 509-892-1203
Email: support@crystalfontz.com
URL: www.crystalfontz.com

CONTENTS

INTRODUCTION

REVISION HISTORY

This Guide

Publication Date: Preliminary Release 2013-11-20

- On our website, we've simplified the way different CFA10036 configurations and optional accessories can be ordered. This *Guide* has been updated to reflect these changes.
- CFA10036 photos were updated to include a heat spreadermade of red aluminum sheet metal. The hardware version for the CFA10036 has not changed.
- In Physical Characteristics (Pg. 14), weight of CFA10036 increased from 9 grams to 19 grams due to addition of heat spreader. Operating temperature range was clarified. The temperature range has not changed.
- Added a section ***SOFTWARE QUICK START GUIDE*** (Pg. 25). For a PC with a Windows operating system, this guide gives detailed step-by-step instructions on how to:
  (1) Install the necessary software developments tools.
  (2) How to use the tools to write a simple program for the CFA10036 to display the text "Hello World" on a connected OLED.

  Some of the early steps for developing a program using a Linux operating system are also included. Although these steps worked successfully for the author of this *Guide*, please note that they did not go through our standard review process.

Publication Date: 2013-06-17

- Added preconfigured username and password to Step 3: Use PuTTY to Open a Terminal.
- Removed description of CFA920 and CFA921. We decided to not to offer these assemblies on our website. CFA920-TS and CFA921-TS with touch screen are still available.
- In CFA10036 Features, deleted reference to Certificate of Compliance: ISO 9001:2008 manufacturer, Product Specifications, RoHS, and REACH:SVHC. The CFA10036 and products that include it do not have a certificate but were manufactured by ISO certified facilities.

Release Date: 2013-04-22
New *Getting Started Guide*.

CFA10036 Hardware

For CFA10036 hardware revision information, see the Part Change Notifications (PCNs) under the "Notices" tab on the CFA10036 web page.

RESOURCES FOR TECHNICAL SUPPORT

Hardware Resources

Crystalfontz manufactures and fully supports the CFA10036 hardware. In addition to the hardware information in this *CFA10036 Getting Started User Guide*, the schematic is available for download as a PDF file under the Docs/Files tab on all of the CFA10036 product web pages.

For additional hardware support, please contact Crystalfontz Technical Support at 888-206-9720 or send an email to support@crystalfontz.com.

Software Resources

For a PC with a Windows operating system, this *Guide* gives detailed step-by-step instructions on how to: (1) Install the necessary software developments tools and (2) How to use the tools to write a simple program for the CFA10036 to display the text “Hello World” on a connected OLED. Some of the early steps for developing a program using a Linux operating system are also included. Although these steps worked successfully for the author of this Guide, please note that they did not go through our standard review process.

For additional software support, this product will be supported by the Linux community by each respective software package maintenance project team.

CFA10036 NOTICES

About Volatility

The CFA10036 contains nonvolatile memory.

Special Terms Applicable for the CFA10036 When Mounted on the Optional CFA10037 Designated as NOT FCC Approved / FCC NOTICE

The CFA10036 mounted on the optional CFA10037 (written here as CFA10036+CFA10037) has NOT been authorized under the rules of the Federal Communications Commission and may not be offered for further sale or lease, or sold or leased, until authorization is obtained. FCC experimental licensing requirements may apply to Buyer's use.

The CFA10036+CFA10037 is made available solely to allow product developers to evaluate Crystalfontz technology and/or software associated with the CFA10036+CFA10037 to determine whether to incorporate such items in a finished product. This is not a finished product and when assembled may not be resold or otherwise marketed unless all required FCC equipment authorizations are first obtained. Operation is subject to the conditions that this CFA10036+CFA10037 not cause harmful interference to licensed radio stations and that this CFA10037 accept harmful interference. Unless the assembled kit is designed to operate under Part 15 or Part 95 of the FCC Rules, the operator of the kit must operate under the authority of an FCC license holder or must secure an experimental authorization under Part 5 of the FCC Rules.

This CFA10036+CFA10037 is not an end product or finished appliance and is not intended or authorized to be integrated into end products as is. Buyer may use the CFA10036+CFA10037 to create prototypes for its own internal experimental use, and may add components, features, software and/or functionality, in the process of designing its own final products. Buyer shall only assign technically qualified electronics experts who are familiar with the dangers and application risks associated with handling electrical mechanical components, subsystems and systems to use the CFA10036+CFA10037, and that such use takes place solely in laboratory/development environments (and not in residential environments). The Non-FCC approved CFA10036+CFA10037 is FOR ENGINEERING DEVELOPMENT, DEMONSTRATION, AND/OR EVALUATION PURPOSES ONLY.

Additional Fine Print

Certain applications using Crystalfontz America, Inc. products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications"). CRYSTALFONTZ AMERICA, INC. PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. Inclusion of Crystalfontz America, Inc. products in such applications is understood to be fully at the risk of the customer. In order to minimize risks associated with customer applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazard. Please contact us if you have any questions concerning potential risk applications.

Crystalfontz America, Inc. assumes no liability for applications assistance, customer product design, software performance, or infringements of patents or services described herein. Nor does Crystalfontz America, Inc. warrant or represent that any license, either express or implied, is granted under any patent right, copyright, or other intellectual property right of Crystalfontz America, Inc. covering or relating to any combination, machine, or process in which our products or services might be or are used.

All specifications in Data Sheets and on our website are, to the best of our knowledge, accurate but not guaranteed. Corrections to specifications are made as any inaccuracies are discovered.

Company and product names mentioned in this publication are trademarks or registered trademarks of their respective owners.

CONVENTIONS USED IN THIS GUIDE

□ **File or folder or name:** Text is in italics. Variable text within a file or folder name is UPPERCASE.
Example: *arm-DATE-arm-none-linux-gnueabi.exe*.

□ **Location or path:** Text is in Courier font.
Example: `C:\Program Files (x86)\CodeSourcery\`

□ **Command to execute in a terminal window:** Text is in Courier font and highlighted in gray.
Example: `CC=arm-none-linux-gnueabi-gcc`↵
The "↵ " indicates press Enter key.

□ **Make a selection from a list or menu:** Text is within quote marks. A menu-within-a-menu is indicated with ">".
Examples: Select "IA32 Windows Installer". Select "File>New".

□ **Click on a button:** Then name of the button is within quote marks.
Example: Click on the "Load" button.

□ **Click on a shortcut:** The name of the shortcut is in italics.
Example: Click on the *Git Bash* shortcut.

□ **Web page heading or subheading:** Text is in italics.
Example: The GitHub web page with the heading *Plans & Pricing* will display.

OVERVIEW OF THE CFA10036

In the past, new hardware designs consisted of choosing a processor, memory, various peripherals, and an operating system. Much of the effort to design a new product was in the routing of the high speed lines for the processor and memory architecture. By using the low-profile high-density ARM9-based CFA10036 SOM (System On Module), all of the high speed design work is done. Shipped installed with a full Linux operating system, it is now easier to take a design concept and implement it quickly and cost effectively.

You have a choice of two powerful Freescale i.MX28 ARM9 processors for the six layer impedance-controlled PCB. The processors are in Freescale's longevity program and availability is assured for a minimum of 15 years. For more information about the two processors, see Compare i.MX283 And i.MX287 Processors (Pg. 12).

Both processors provide the CFA10036 with *lots* of GPIO. The CFA10036 connects to a standard, widely available 200-pin SODIMM socket (available from Digi-Key).

Most other SOM's or SBC's (Single Board Computers) contain flash memory soldered as chips onto the SOM PCB for nonvolatile storage. Using the CFA10036's microSD card socket eliminates the need for on-board flash memory and the software tools to support it. The CFA10036 is shipped with one card inserted in the microSD card socket. Using an SD card opens up fantastic on-board user data and logging possibilities.

For easy status display and debugging, we offer the optional small-but-pixel-dense128x32 OLED, the CFAL12832D-B.

Power can be supplied through the micro USB connector, the DUART +5v interface, or by an external lithium-ion battery (Lion). All other voltages (+3.3v, +1.8v, and +1.5v) are internally generated.

Because a full Linux mainline kernel is already ported to the CFA10036, you can devote your resources to applications in the languages of your choice. All software is open source, will is maintained by the Linux community, and will be available through GitHub here: github.com/crystalfontz/cfa_10036_kernel.

Rugged and easy to embed, the CFA10036 is an ideal platform for production runs where costly components such as the processor and memory do not need to be mounted on the customer hardware. The Linux environment on the CFA10036 has support for all major hardware communications including I^{2}C, SPI, UART, Ethernet, WiFi (USB), and CAN, rotary encoder, and multiple LCD support. Custom code does not have to be created for customer hardware requiring these interfaces.

If your project requires other functionality, you can build the Linux kernel using Linux or Windows. See APPENDIX A: HOW TO INSTALL UBUNTU ON WINDOWS 8 USING VMWARE PLAYER (Pg. 63) and APPENDIX B: HOW TO BUILD THE LINUX DISK IMAGE (Pg. 74).

Inexpensive carrier boards and development boards can be purchased (or manufactured by you) and tested using a single CFA10036. For example, you may want to use a CFA10036 with a CFA10037 development board in a prototype design using I^{2}C interface. You could use the same CFA10036 with a second CFA10037 development board in a prototype design using SPI interface. See ADD-ON PRODUCTS FOR THE CFA10036 (Pg. 20).

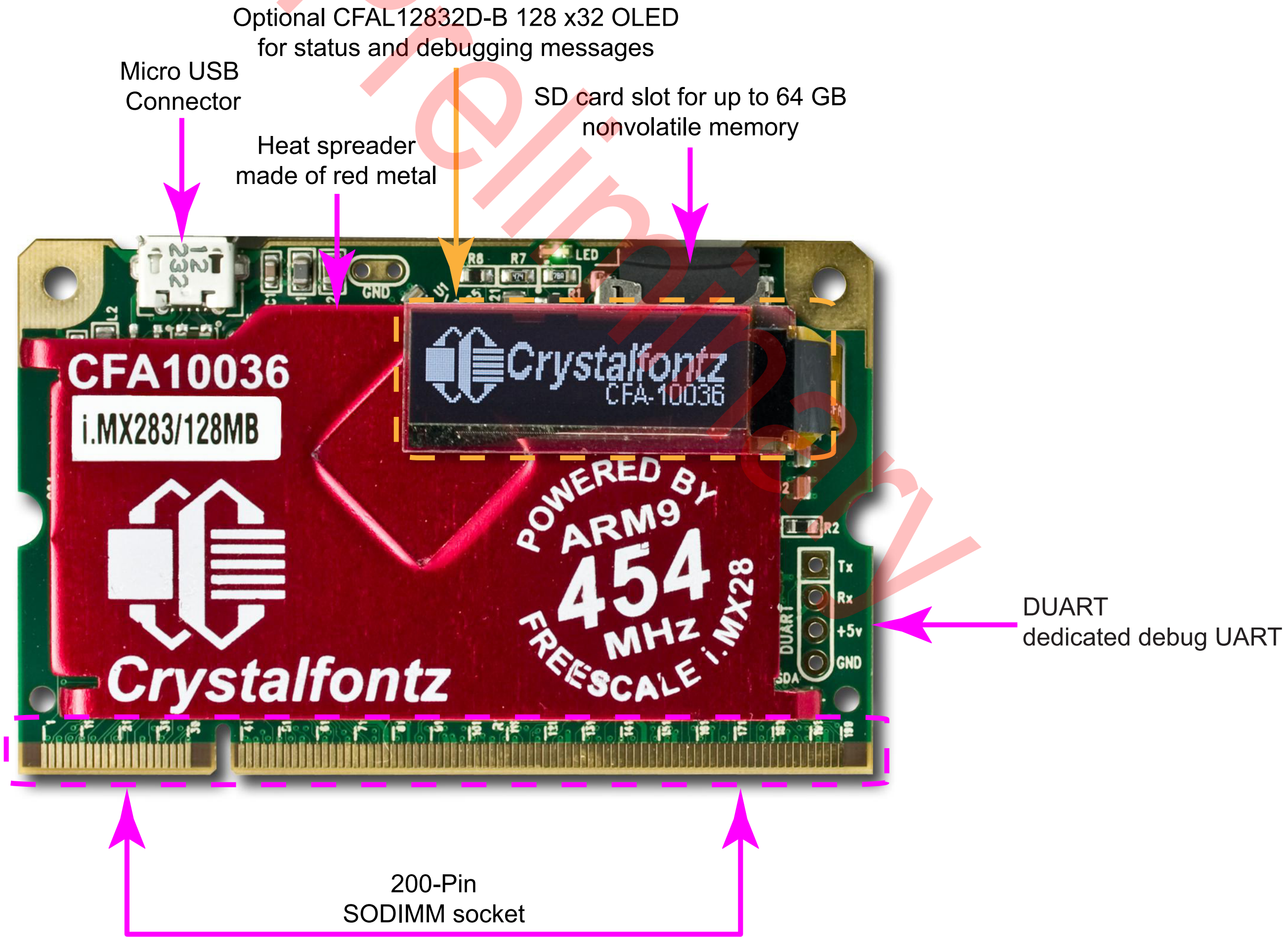
Examples of designs that could benefit from a CFA10036:

- Home automation systems that require ZigBee communications can be easily added using the I^{2}C interface.
- Alarm systems where a large number of GPIO pins for sense and control are required.
- The CFA10036 can easily monitor several environmental sensors (temperature, humidity, etc), log the data, and make that data available over the Internet. This is the core concept behind IoT (Internet of Things). Starting with a CFA10036 makes it very easy to make *your* thing part of the IoT.

MAIN FEATURES

The CFA10036 has many standard features that make it an excellent core module for designing your next project. Below is a photo showing the features for the CFA10036-1. The CFA10036-1 has the i.MX283 processor. The photo below includes an optional preinstalled OLED.

HARDWARE FEATURES



Features for the CFA10036 line include:

❑ Two choices of Freescale i.MX28 processors:

1. The *i.MX283* processor with*128 MB* DDR2 SDRAM running at 454Mhz with 91 GPIOs.
2. The *i.MX287* processor with *256 MB* DDR2 SDRAM running at 454Mhz with 126 GPIOs.

❑ 200-pin SODIMM brings out all processor functionality (with the exception of DRAM interface) to allow complete customization and configuration of the CFA10036.

❑ An optional small-but-pixel dense CFAL12832D-B 128x32 OLED can be mounted on the CFA10036 for status and debugging messages.

❑ On-board micro-AB receptacle connector for USB 2.0 OTG makes communications simple. USB 2.0 host controller with PHY is included.

❑ MicroSD/microSDHC/microSDXC support for up to 64 GB of nonvolatile storage.

❑ On-board DUART connectivity for debugging and early board bring up. Dedicated Rx/Tx debug UART port with 0.1" centers. You can connect a USB633 TTL-to-USB converter to the dedicated DUART.

❑ Four ways to connect power:

1. Directly through the micro USB connector.
2. +5 volts to DUART connection.
3. +5 volts through the 200-pin SODIMM socket.
4. Lithium-ion (Lion) battery through 200-pin SODIMM socket.

❑ Dense BGA package: 289-ball BGA 17x17 grid on 0.8mm pitch.

❑ The CFA10036 and products that include it do not have a certificate but were manufactured by ISO certified facilities.

SOFTWARE FEATURES

❑ Complete open source ARM Linux kernel. All CFA10036 specific features are checked in to the mainline Linux kernel and will be supported by the Linux community.

❑ Board support packages for the CFA10036 and development boards (see ADD-ON PRODUCTS FOR THE CFA10036 (Pg. 20)).

❑ Full support for four sizes of touchscreen TFT displays when purchased as an assembled product.

❑ Driver support for peripherals include rotary encoders, ADC converters, DACs, WiFi, Ethernet, keypad, and others.

FEATURES OF FREESCALE I.MX28 PROCESSORS

The following lists the features of the i.MX28 processor line. (Note: not all features are available for both processors choices, the i.MX283 and the i.MX287.) See the table below this section for a list of the differences between the i.MX283 and the i.MX287 processors. For additional information, see the Freescale i.MX28 Data Sheet.

❑ ARM926EJ-S CPU running at 454 MHz.

❑ 16-Kbyte instruction cache and 32-Kbyte data cache.

❑ ARM embedded trace macrocell (CoreSightTM ETM9TM).

❑ Parallel JTAG interface.

❑ 128 Kbytes of integrated low-power on-chip SRAM.

❑ 128 Kbytes of integrated mask-programmable on-chip ROM.

❑ 1280 bits of on-chip one-time-programmable (OCOTP) ROM.

❑ 16-bit mobile DDR (mDDR) (1.8 V), DDR2 (1.8 V) and LV-DDR2 (1.5 V), up to 205 MHz DDR clock frequency with voltage overdrive.

❑ Four synchronous serial ports (SSP) for SDIO/MMC/MS/SPI: SSP0, SSP1, SSP2, and SSP3. SSP0 and SSP1 can support three modes,1-bit, 4-bit, and 8-bit, whereas SSP2 and SSP3 can support only 1-bit and 4-bit modes.

❑ Ethernet MAC

  - ● With the *i.MX283* processor, 10/100-Mbps IEEE 802.3 Ethernet MAC is supported.
  - ● With the *i.MX287* processor,10/100-Mbps 802.3 Ethernet MAC is 2x is supported.
- ❑ Single 10/100 Ethernet with GMII/RMII or Dual 10/100 Ethernet with RMII interface.
- ❑ Supporting IEEE Std 1588TM-compatible hardware timestamp.
- ❑ Supporting 50-MHz/25-MHz clock output for external Ethernet PHY.
- ❑ Two 2.0B protocol-compatible Controller Area Network (CAN) interfaces.
- ❑ One USB2.0 OTG device/host controller and PHY.
- ❑ One USB2.0 host controller and PHY.
- ❑ LCD controller, up to 24-bit RGB (DOTCK) modes and 24-bit system-mode.
- ❑ Pixel-processing pipeline (PXP) supports full path from color-space conversion, scaling, alpha-blending to rotation without intermediate memory access.
- ❑ SPDIF transmitter.
- ❑ Dual serial audio interface (SAIF) to support full-duplex transmit and receive operations; each SAIF supports three stereo pairs.
- ❑ Five application Universal Asynchronous Receiver-Transmitters (UART), up to 3.25 Mbps with hardware flow control.
- ❑ One debug UART operating at up to 115 Kb/s using programmed I/O.
- ❑ Two I2C master/slave interfaces, up to 400 kbps.
- ❑ Four 32-bit timers and a rotary decoder.
- ❑ Eight Pulse Width Modulators (PWMs).
- ❑ Real-time clock (RTC) with 32KHz crystal (requires continuous power).
- ❑ GPIO with interrupt capability.
- ❑ Power Management Unit (PMU) supports a triple output DC-DC switching converter, multiple linear regulators, battery charger, and detector.
- ❑ 16-channel Low-Resolution A/D Converter (LRADC). There are 16 physical channels but they can only be mapped to 8 virtual channels at a time.
- ❑ Single channel High Speed A/D Converter (HSADC), up to 2 Msps data rate.
- ❑ 4/5-wire touchscreen controller.
- ❑ Up to 8X8 keypad matrix with button-detect circuit.
- ❑ Security Features
  - ● Read-only unique ID for Digital Rights Management (DRM) algorithms.
  - ● Secure boot using 128-bit AES hardware decryption.
  - ● SHA-1 and SHA256 hashing hardware.
  - ● High assurance boot (HAB4).

COMPARE I.MX283 AND I.MX287 PROCESSORS

The following table lists the functional differences between the two processor choices.

Function	i.MX283	i.MX287
DDR2 RAM	128 MB	256 MB
GPIOs	91	126
Application UART	x5	x5
Debug UART	x1	x1
CAN	—	x2
Ethernet	x1	x2
High-speed ADC	x1	x1
L2 Switch	—	Yes
LCD Interface	Yes	Yes
LRADC	x8	x8
PWM	x8	x8
S/PDIF Tx	—	Yes
SD/SDIO/MMC	x4	x4
Security	Yes	Yes
SPI	x4	x4
Touch Screen	Yes	Yes
USB 2.0	OTG HS with HS PHY x1	OTG HS with HS PHY
	HS Host with HS PHY x1	HS Host with HS PHY x1
JTAG	Yes	Yes
For more information, see *the Freescale i.MX28 Data Sheet*.		

HARDWARE DESIGN INFORMATION

The hardware design schematic can be downloaded on our website under the Doc/Files tab as a PDF file.

OUTLINE DRAWING

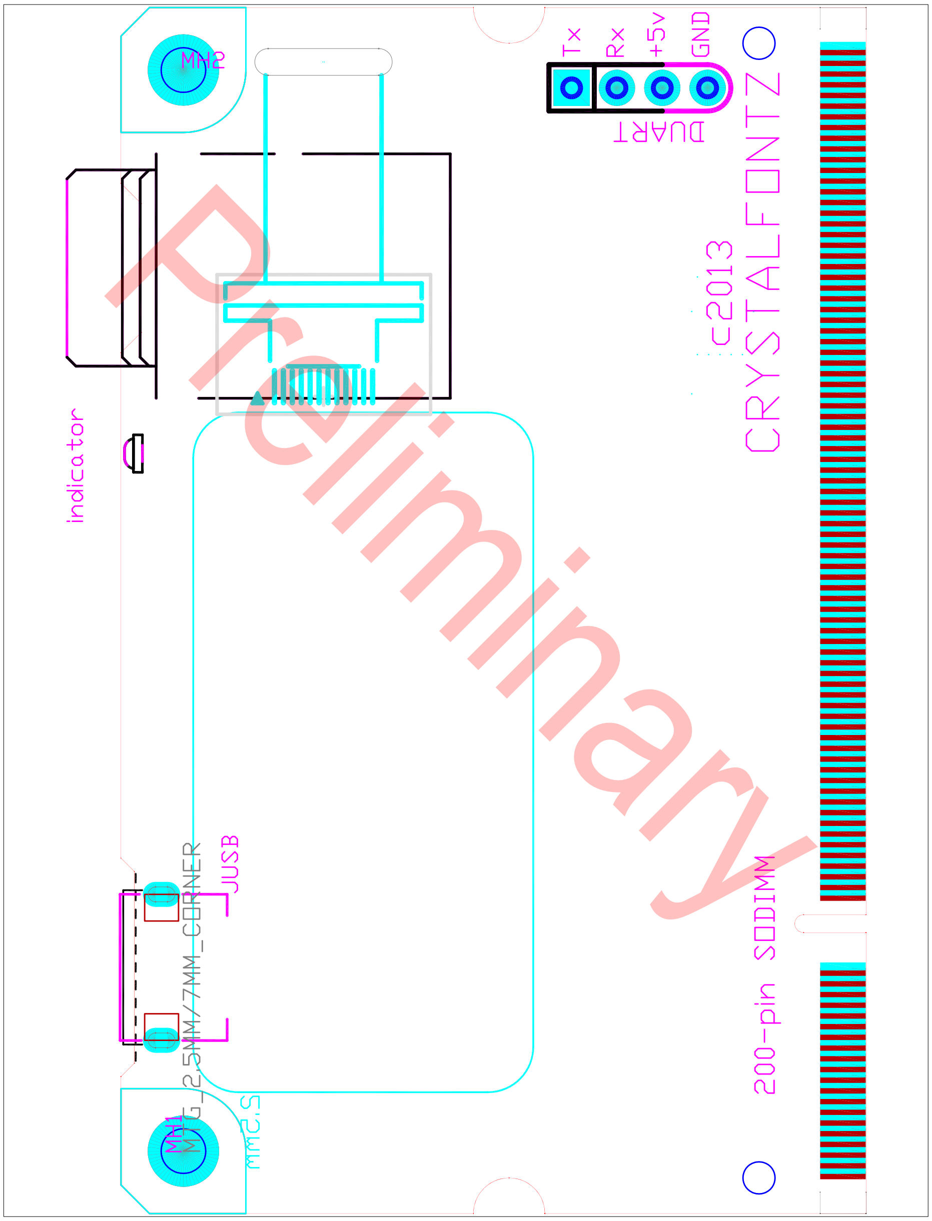


Figure 1. Outline Drawing

PHYSICAL CHARACTERISTICS

CFA10036 Specifications Without Optional Add On Products	
Overall Dimensions (without optional OLED)	67.60 (W) x 71.80 (H) millimeters
Weight	19 grams
Operating Temperature	-20°C to +70°C
Storage Temperature	-30°C to +80°C
Humidity (RH noncondensing)	0%

VIBRATION

Test conditions:

- GR-63-CORE 5.4.2, Office Vibration, Alternative Test: 5-100-5 Hz at 1.0 g with a sweep rate of .25
- Octave/minute, 35 minutes per axis.
- MIL-STD 810F, Figure 514C-17, Random: 1 hour per axis.
- MIL-STD 810F, Figure 514C-18, Sine: 1 hour per axis.

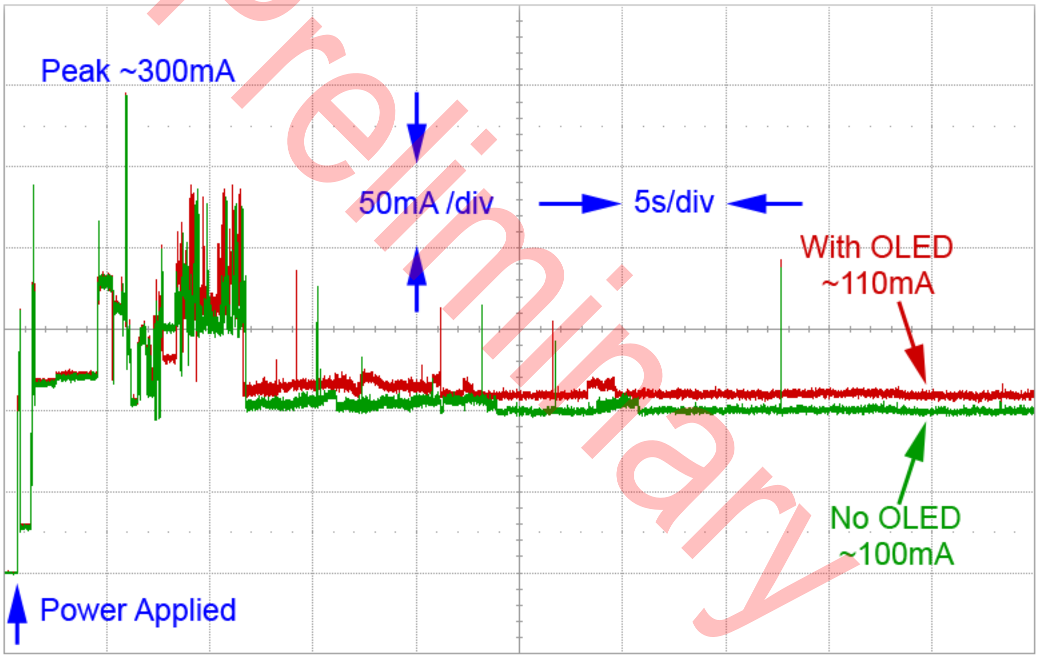
For details see APPENDIX D: VIBRATION TEST REPORT (Pg. 77).

ELECTRICAL SPECIFICATIONS

DC CHARACTERISTICS

Electrical Specification	Without Optional CFAL12832D-B	With Optional CFAL12832D-B
Typical current consumption at +5.0v	100 mA	110 mA
Typical peak current draw during boot process	350 mA	350 mA
Constant busy loop	120 mA	130 mA

CFA-10036 Typical Boot Current Profile (5v supply)



GPIO CURRENT LIMITS

TYPICAL GPIO CURRENT LIMITS	
Sink	TBD mA
Source	TBD mA

REFERENCE FOR PIN FUNCTIONS

Freescale has a tool to assist in assigning pin multiplex configurations. The tool will identify “on the fly” any assignment collisions or multiplexing issues. To download, go here and look for the *Interactive i.MX Pin Mux Tool*.

This tool is great for documenting your design. Design configurations can be saved and exported for future use in schematics or software source code as supplementary documentation of a system.

CARE AND HANDLING PRECAUTIONS

For optimum operation of the CFA10036 and to prolong its life, please follow the precautions described below. If you have a CFAL12832D-B or other display module mounted on the CFA10036, please be sure to read the *Care and Handling Precautions* in the module's Data Sheet found on this product's web page under the Doc/Files tab.

HANDLING CAUTION FOR PRODUCTS SHIPPED IN TRAYS

If you receive products packed in trays, handle trays carefully by supporting the entire tray. Trays were made to immobilize the modules inside their packing carton. Trays are not designed to be rigid. Do not carry trays by their edges; trays and modules may be damaged.

ESD (ELECTRO-STATIC DISCHARGE) SPECIFICATIONS

The circuitry is industry standard CMOS logic and is susceptible to ESD damage. Please use industry standard antistatic precautions as you would for any other static sensitive devices such as expansion cards, motherboards, or integrated circuits. Ground your body, work surfaces, and equipment.

DESIGN AND MOUNTING

- Do not disassemble or modify.
- Solder only to the I/O terminals.
- Do not reverse polarity to the power supply connections. Reversing polarity will immediately ruin the product.

AVOID SHOCK, IMPACT, TORQUE, OR TENSION

- Do not expose to strong mechanical shock, impact, torque, or tension.
- Do not drop, toss, bend, or twist.
- Do not place weight or pressure on the product.

OPERATION

- Your circuit should be designed to protect the product from ESD and power supply transients.
- Observe the operating temperature limitations: a minimum of -20°C to a maximum of +70°C noncondensing with minimal fluctuation. Operation outside of these limits may shorten life and/or harm product.
- Operate away from dust, moisture, and direct sunlight.

STORAGE AND RECYCLING

- Store in an ESD-approved container away from dust, moisture, and direct sunlight with humidity less than 90% noncondensing.
- Observe the storage temperature limitations: a minimum of -30°C minimum to +80°C noncondensing maximum with minimal fluctuations. Rapid temperature changes can cause moisture to form, resulting in permanent damage.
- Do not allow weight to be placed on the products while they are in storage.
- To discard, please recycle your products at an approved facility.

LONGEVITY (EOL / REPLACEMENT POLICY)

Crystalfontz is committed to making the CFA10036 available for as long as possible. Occasionally, a supplier discontinues a component, or a process used to make the module becomes obsolete, or the process moves to a more modern manufacturing line. In order to continue making the module, we will do our best to find an acceptable replacement part or process which will make the “replacement” fit, form, and function compatible with its predecessor.

We recognize that discontinuing a product may cause problems for some customers. However, rapidly changing technologies, component availability, or low customer order levels may force us to discontinue (“End of Life”, EOL) a product. For example, we must occasionally discontinue a product when a supplier discontinues a component or a manufacturing process becomes obsolete. When we discontinue a product, we will do our best to find an acceptable replacement module with the same fit, form, and function.

In most situations, you will not notice a difference when comparing a “fit, form, and function” replacement module to the discontinued module it replaces. However, sometimes a change in component or process for the replacement module results in a slight variation, perhaps an improvement, over the previous design.

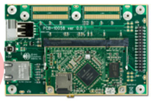
Although the replacement product is still within the stated Data Sheet specifications and tolerances of the discontinued module, changes may require modification to your circuit and/or firmware. Possible changes include:

- *Processor.* We may switch to a different processor if the current processor becomes unavailable.
- *RAM.* The configuration of RAM may be different as the availability of RAM supplies change.
- *Linux Operating System.* We may freeze the Linux version to a given release or support new Linux versions as they become available.

Please understand that we avoid changing a product whenever possible; we only discontinue a product if we have no other option. We publish Part Change Notices (PCN) as soon as possible. PCNs can be found under the “Notices” tab on the CFA10036 web page.

ASSEMBLED PRODUCTS WITH CFA10036

Assembled products in the table below have a CFA10036 SOM (your choice of the i.MX283 or i.MX287 processor), a mother board, and a TFT.

Product	CFA10036 Installed On *Bottom* Of Mother Board	TFT Installed On *Top* Of Mother Board	Active Area
CFA920-TS			4.3-inch diagonal 480x800
CFA921-TS			5.0-inch diagonal 800x480

MOTHER BOARDS IN ASSEMBLED PRODUCTS

Features:

- 10/100 Ethernet port.
- USB port.
- 128 MB DDR2 RAM.
- µSD up to 64 GB.
- 24 GPIO on easy-access 0.1” center header.
- Up to 108 GPIO on a 1 mm compression connector.

Note: Mother boards are not available as a separate product.

TFT DISPLAYS IN ASSEMBLED PRODUCTS

The TFTs are 262K full color with a 4-wire resistive touchscreen.

Note: TFT warranty is void if you attempt to remove the preinstalled TFT from the mother board.

CUSTOMIZE ASSEMBLED PRODUCTS

When you go to the web page for the CFA920-TS or CFA921-TS and click on the “Customize and Add to Cart” button, you will see a list of options. You can choose to have an OLED installed on the CFA10036 for status messages and debugging.

Additional options are to add a power supply and a USB cable to your order. For descriptions of these add-on products, see in the ADD-ON PRODUCTS FOR THE CFA10036 (Pg. 20) section below.

ADD-ON PRODUCTS FOR THE CFA10036

If you click on the individual website page links for these CFA10036 products, in the right panel of their web pages you will see that some products have an "Add to Cart' button while other products have a "Customize and Add to Cart" button.

When you click on a "Customize and Add to Cart" button, you will see a list of options that includes add-on products. The photo below is an example of the CFA10036 with add-on products.

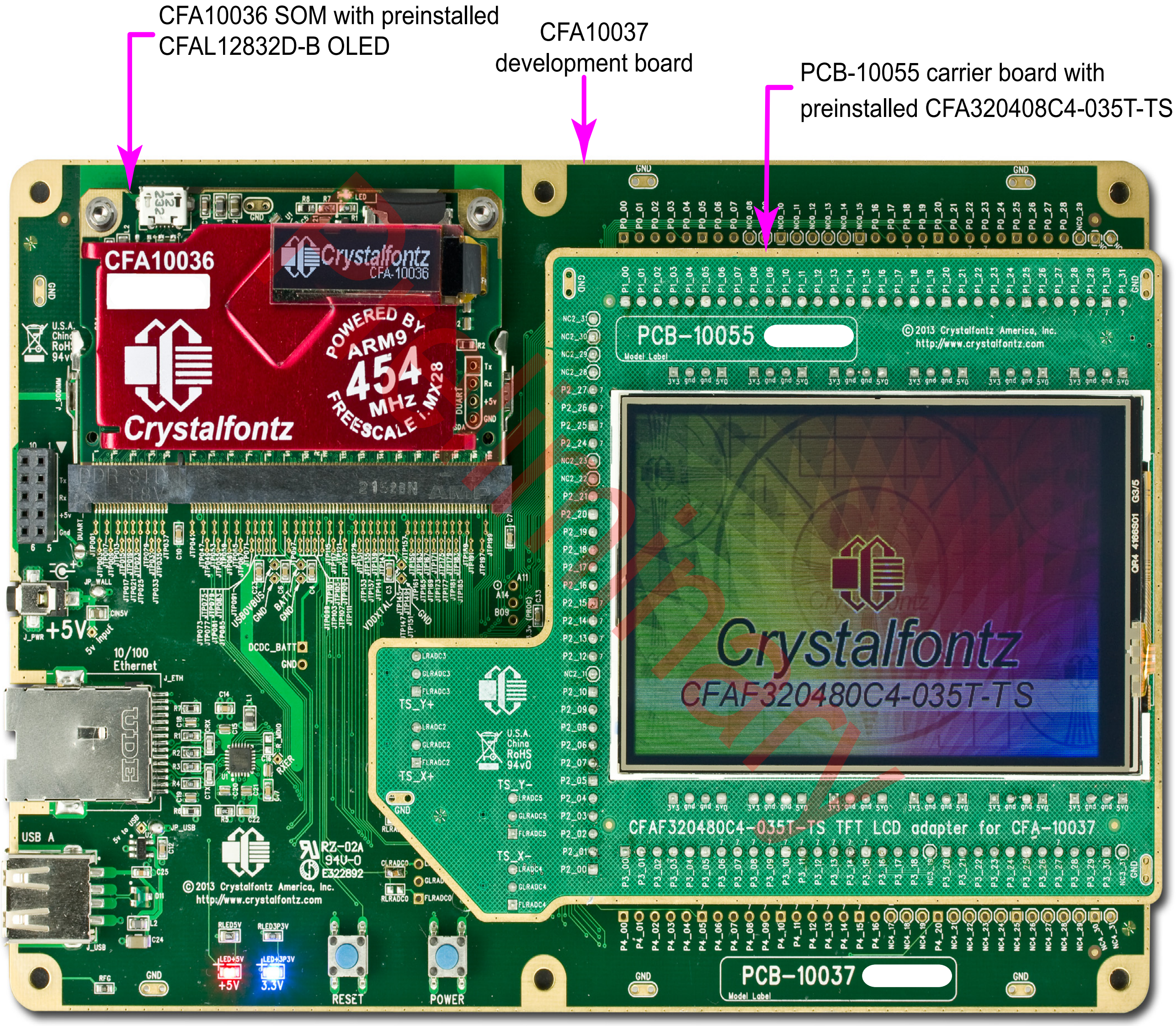

CFA10036 SOM with preinstalled
CFAL12832D-B OLED
CFA10037
development board
PCB-10055 carrier board with
preinstalled CFA320408C4-035T-TS

Figure 2. Example of A CFA10036 With Add-On Products

If you looked at the assembly in the previous photo on edge, this is what you would see, from top to bottom:

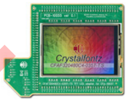
	CFA10036 Photo With Add-On Products	
	Installed On Left Side	Installed On Right Side
Top	CFAL12832D-B OLED	CFAF320480C4-035T-TS
Center	CFA10036-2 SOM	CFA10055 Carrier Board
Bottom	CFA10037 Development Board	

Below are descriptions of the major add-on products.

CARRIER BOARDS WITH PREINSTALLED TFT DISPLAYS

Carrier boards have a 262K full color TFT display preinstalled on them. The carrier board+TFT is designed to work with the CFA10036 SOM+CFA10037 development board, purchased separately.

These two carrier boards with TFT displays are available:

Product	TFT Installed On Top Of Carrier Board		Active Area
CFA10055-8	CFAF320480C4-035T-TS with 4-wire resistive touchscreen		3.5-inch diagonal 320x480
CFA10056-4	CFAF480800FT2-040T without touchscreen		4.0-inch diagonal 480x800

Note: TFT warranty is void if you attempt to remove the TFT from the carrier board.

OLED

The CFAL12832D-B 128x32 OLED graphic module displays light (near-white) characters on a dark (near-black) background. Less than 0.5-inch high (11.5 mm), the CFAL12832D-B can be installed onto the CFA10036 with a ZIF connector. The OLED is useful for status messages and debugging.

Note: If you order this OLED separately from the CFA10036 (at a later time), the OLED on the CFA10036 will not by default display anything other than the 'random' bits from the display RAM on the CFAL12832D-B (uninitialized display RAM).

DEVELOPMENT BOARD

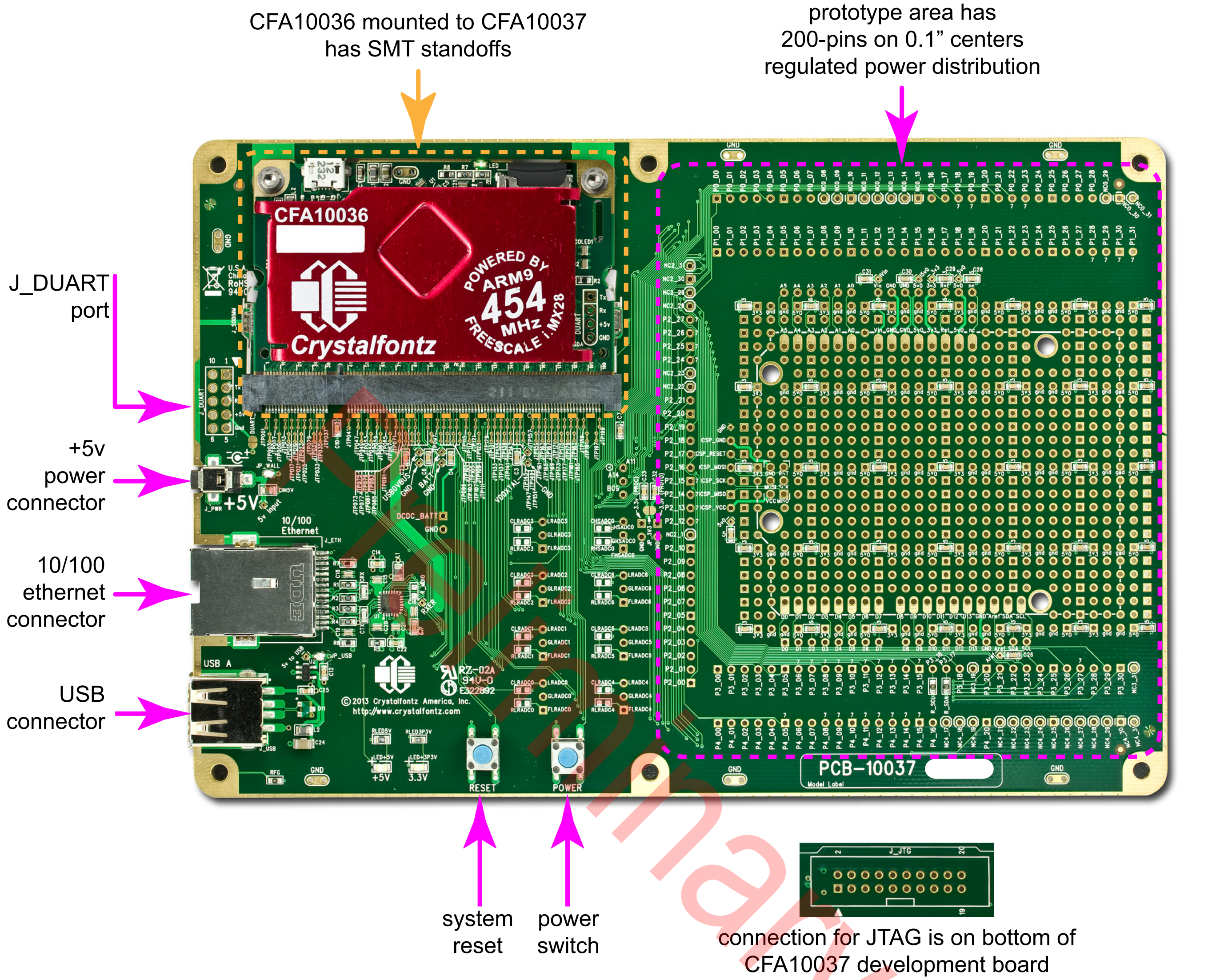


Figure 3. Parts On The CFA10037 Development Board

The CFA10036 mounted on the optional CFA10037 is ideal for rapid prototyping. The CFA10037 gives access to all the port pins of the CFA10036, and by extension the port pins of the i.MX28 processor. It includes an Ethernet PHY connector and USB A connector. Signals are brought out for connection to a compatible JTAG debugger.

With the 200-pin SODIMM form factor, all processor pins with the exception of memory are brought out for your use in a prototype area with solid power distribution. Using 0.1" center headers and socketed hook-up wire, you can connect any of the pins on the CFA10036 to any location on the CFA10037's prototype area.

The CFA10036 connects to the CFA10037 via its JDEC MO-224E SODIMM connector. To get the mechanical stability expected in embedded systems, we slightly extended the SODIMM form factor and added two mounting holes. These mate perfectly with blind threaded SMT standoffs that hold the CFA10036 securely to the CFA10037.

A set of holes line up with shields designed for an Arduino Uno R3.

With the CFA10036 mounted on the optional CFA10037, you can download your applications via TCP/IP through the CFA10037's USB connector using a WiFi adapter dongle.

USB A TO MICRO B CABLE

The WR-USB-Y27 cable has two different types of USB connectors, one smaller than the other. Connect the cable's smaller micro B USB female connector to the CFA10036's micro B male USB connector. About 1.925 meters (6 feet, 3.75-inches). Connect the cable's larger USB-A female connector to your PC's USB-A connector. This cable can be ordered separately.

TTL TO USB CONVERTER

The WR-DBG-Y42 kit includes:

- USB633 PCB. This small PCB can mount on the CFA10036's DUART for debugging and early board bring up. It is a dedicated Rx/Tx debug UART port with 0.1" centers. You can connect a USB633 TTL-to-USB converter to the dedicated DUART.
- WR-USB-Y03 cable. This cable has two different types of USB connectors, one smaller than the other. Connect the cable's smaller 2 mm female USB connector to the module's USB connector. Connect the cable's larger USB-A female connector to host's USB-A connector. About 1.94 meters (6 feet, 4.35 inches). This cable can also be ordered separately.

PROTOTYPE KIT WITH JUMPERS

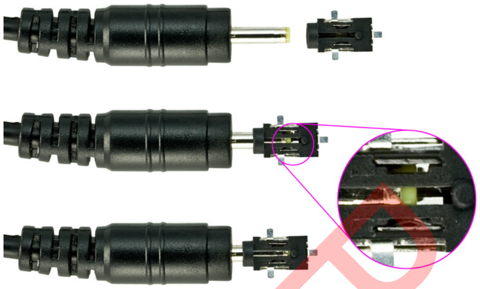
The prototype kit (CFA10037-PROTO) has:

- Female-to-female jumper wires, useful to connect to the CFA10037 prototype area headers. This kit has five packages of twenty for both of these jumpers, a total of 200 jumpers in assorted colors.
  - *WR-JMPY-40*: About six inches long (shown in photo above).
  - *WR-JMPY-41*: About 3.70 inches long.
- Five 36-pin breakaway 0.100" single row pin headers (180 pins).

This kit is sold only as an add-on product when you order the CFA10037. The kit cannot be ordered separately. However, if you click on the blue jumper wire links above, you can order the jumper wires separately.

POWER SUPPLY

The CFA10040PWR is a110 VAC +5v wall power supply that can be used to power the CFA10037 development board. Cord length is about 63 inches.



Note:
The mating connector for the CFA10040PWR has a cable retention feature. Insert the power cable into the connector until there is about 2 mm (or ~3/16") between the connector and the larger part of the cable end. If you insert only until you meet resistance, the power supply may not be fully connected.

SOFTWARE QUICK START GUIDE

The CFA10036 is shipped preinstalled with a full Linux operating system. It can be programmed in many languages, both compiled languages and interpreted languages. Compiled languages such as C and C++ can be used to build applications for the CFA10036 using a host build environment. Interpreted languages such as Java, Perl, and Python do not need to be compiled.

If you have a Windows Operating System: Below are simple step-by-step instructions to install development tools and write a simple "Hello World" program to display on the optional OLED (CFAL12832D-B) mounted to the CFA10036. Please note that although the procedure is simple, it does take time to download and install all of the development tools.

If you have a Linux Operating System: We think that most Linux users are unlikely to need step-by-step instructions on writing and displaying a program on the optional OLED (CFAL12832D-B) mounted to the CFA10036. However, we have included some preliminary instructions to install the development tools.

WINDOWS: DEVELOPMENT TOOLS INSTALLATION

> Note
>
> For the Windows instructions below, we used *Windows 7*. If you use a different operating system (for example, *Windows XP*), the steps below will be similar.

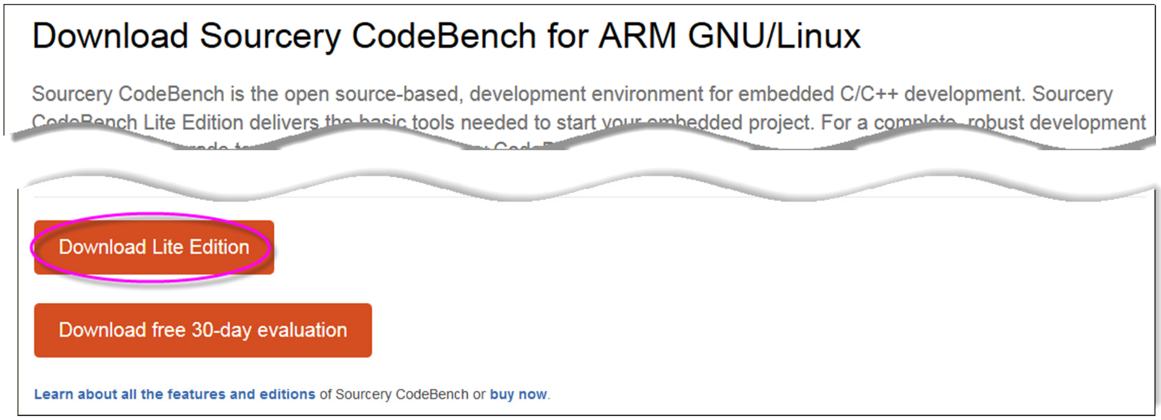
CROSS COMPILER

You will need a cross compiler program. We used the free *Sourcery CodeBench Lite arm-2013.05-24-arm-none-linux-gnueabi.exe* from Mentor Graphics.

> Note
>
> Mentor Graphics has several versions of Sourcery CodeBench Lite with slightly different names. To ensure you have the program that works with these instructions, carefully follow the steps below.

1. Go to https://sourcery.mentor.com/GNUToolchain/release2450.

2. The web page *Download Sourcery CodeBench for ARM GNU/Linux* will display. Click on the orange button “Download Lite Edition” near the bottom of the page.

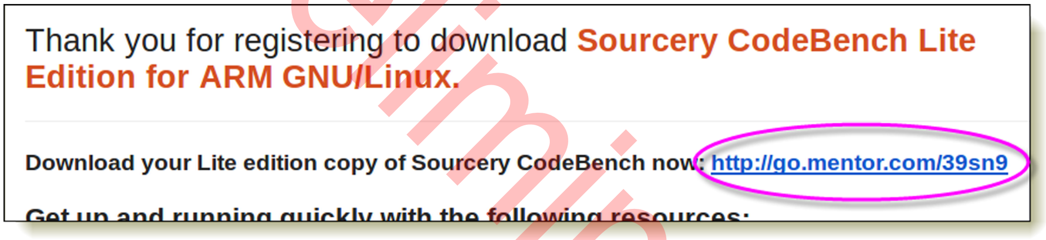


Download Sourcery CodeBench for ARM GNU/Linux

Sourcery CodeBench is the open source-based, development environment for embedded C/C++ development. Sourcery CodeBench Lite Edition delivers the basic tools needed to start your embedded project. For a complete, robust development

Learn about all the features and editions of Sourcery CodeBench or buy now.

3. If you are not already signed in to Mentor Graphics, a web page will open where you will be prompted to create and activate a free account.
4. Create and sign in to your account. A web page will display with the heading *Check your E-mail for Download Link*.
5. Find and open the email from *sourcery_support*. The email acknowledges your registration and includes a unique link to download the program.



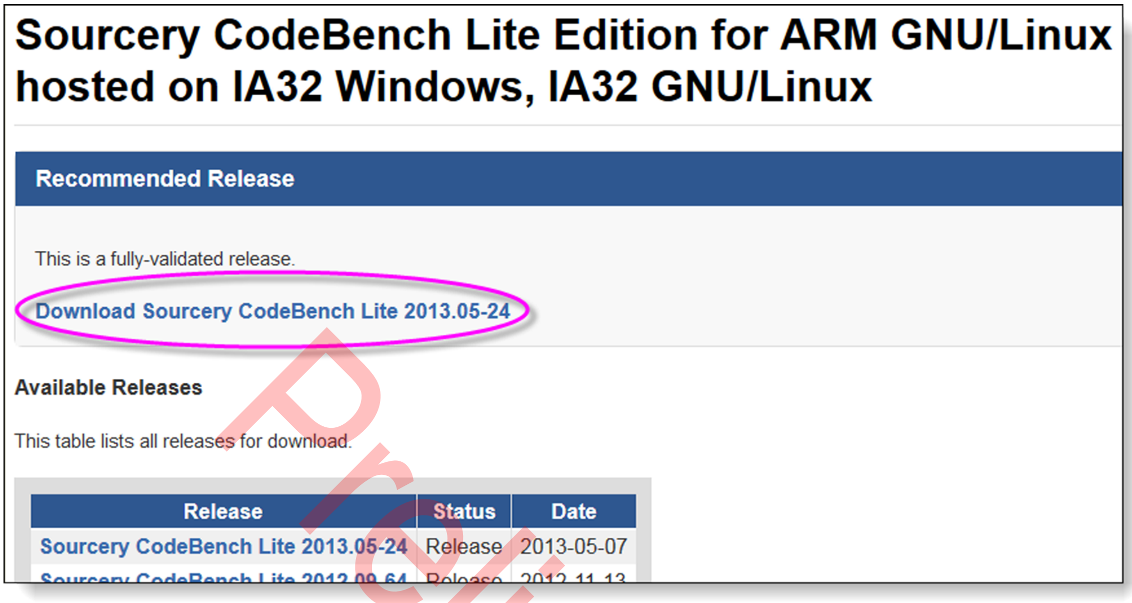
Thank you for registering to download **Sourcery CodeBench Lite Edition for ARM GNU/Linux.**

Download your Lite edition copy of Sourcery CodeBench now: http://go.mentor.com/39sn9

Get up and running quickly with the following resources:

6. A web page with the heading *Sourcery CodeBench Lite Edition for ARM GNU/Linux hosted on IA32 Windows, IA32 GNU/Linux* will display. Note: this page may display or it may be omitted and “step 7.” will display instead.

   You will see a list of downloads. Under the *Recommended Release* subheading, click on the “Download Sourcery CodeBench Lite 2013-.05-24”.



Sourcery CodeBench Lite Edition for ARM GNU/Linux hosted on IA32 Windows, IA32 GNU/Linux

Recommended Release

This is a fully-validated release.

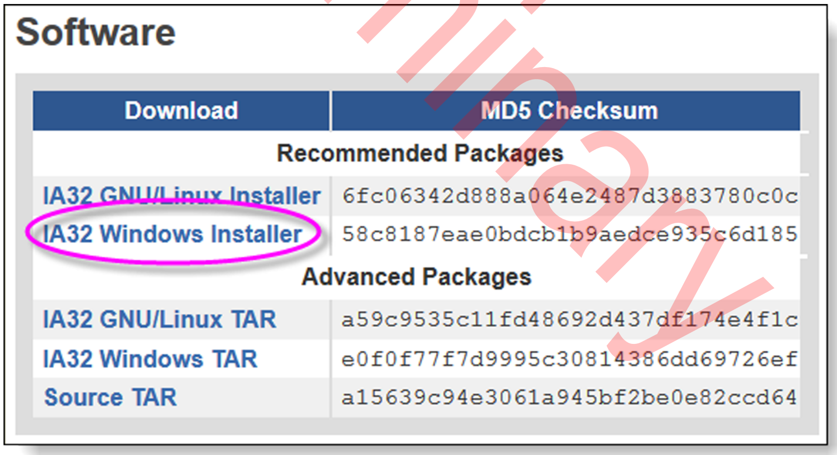
Download Sourcery CodeBench Lite 2013.05-24

Available Releases

This table lists all releases for download.

Release	Status	Date
Sourcery CodeBench Lite 2013.05-24	Release	2013-05-07
Sourcery CodeBench Lite 2012.09-64	Release	2012-11-13

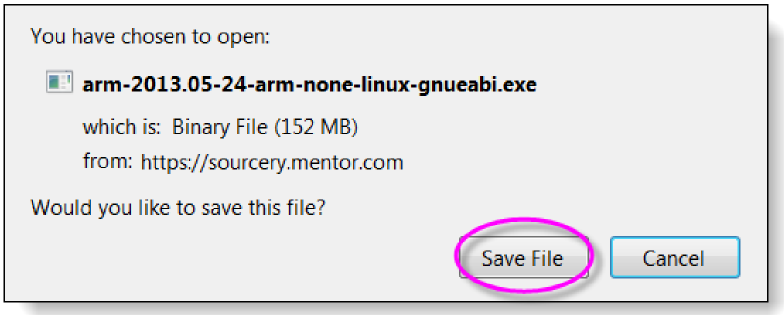
7. A page with the heading *Sourcery CodeBench Lite 2013.05-24* will display. Under the *Software* heading, you will see a list of downloads. Select “IA32 Windows Installer”.



Software

Download	MD5 Checksum
Recommended Packages	
IA32 GNU/Linux Installer	6fc06342d888a064e2487d3883780c0c
IA32 Windows Installer	58c8187eae0bdcb1b9aedce935c6d185
Advanced Packages	
IA32 GNU/Linux TAR	a59c9535c11fd48692d437df174e4f1c
IA32 Windows TAR	e0f0f77f7d9995c30814386dd69726ef
Source TAR	a15639c94e3061a945bf2be0e82ccd64

8. You will be prompted to save an .exe file (*arm-DATE-arm-none-linuxgnueabi.exe*). Save the file to your computer.



You have chosen to open:

arm-2013.05-24-arm-none-linux-gnueabi.exe

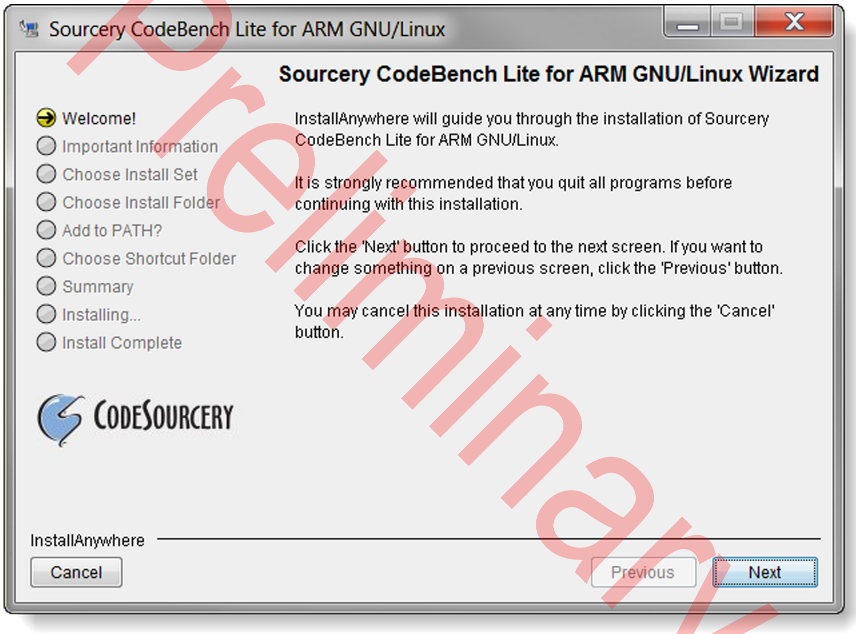
which is: Binary File (152 MB)

from: https://sourcery.mentor.com

Would you like to save this file?

Note the location where you saved the file. Example location: `C:\Users\YOUR USER NAME\Downloads`.

9. Go to the location where you downloaded the *arm-DATE-arm-none-linuxgnueabi.exe* file and double click on it. The installation wizard *Sourcery CodeBenchLite for ARM GNU/Linux Wizard* will open.



Sourcery CodeBench Lite for ARM GNU/Linux

Sourcery CodeBench Lite for ARM GNU/Linux Wizard

InstallAnywhere will guide you through the installation of Sourcery CodeBench Lite for ARM GNU/Linux.

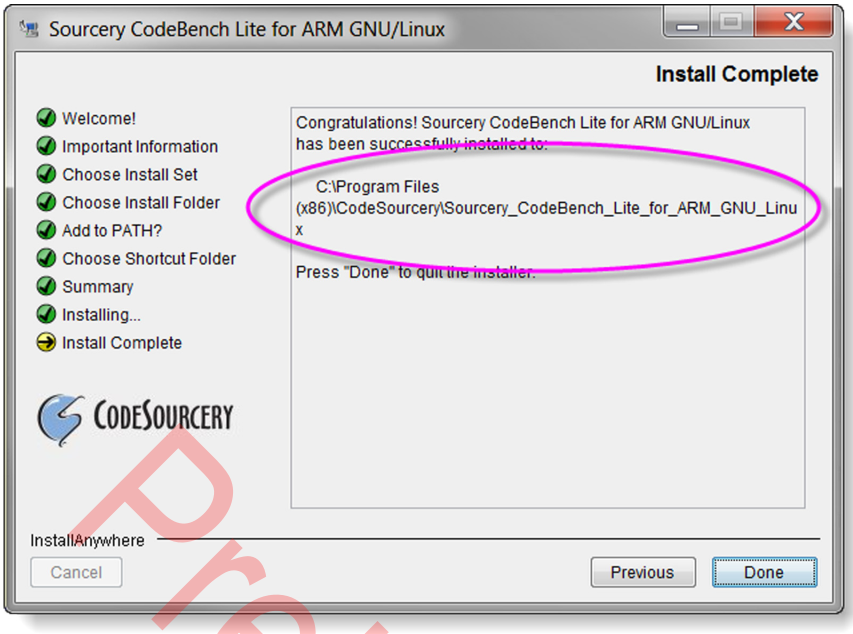
It is strongly recommended that you quit all programs before continuing with this installation.

Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

10. Follow the steps in the wizard's instructions for a *Typical* installation. The default options are fine.

11. Look for the program's location shown on the wizard's final window with the heading *Install Complete.* The program files will be installed in a folder named *CodeSourcery* within your computer's *Program Files(x86)* folder.



Congratulations! Sourcery CodeBench Lite for ARM GNU/Linux has been successfully installed to:

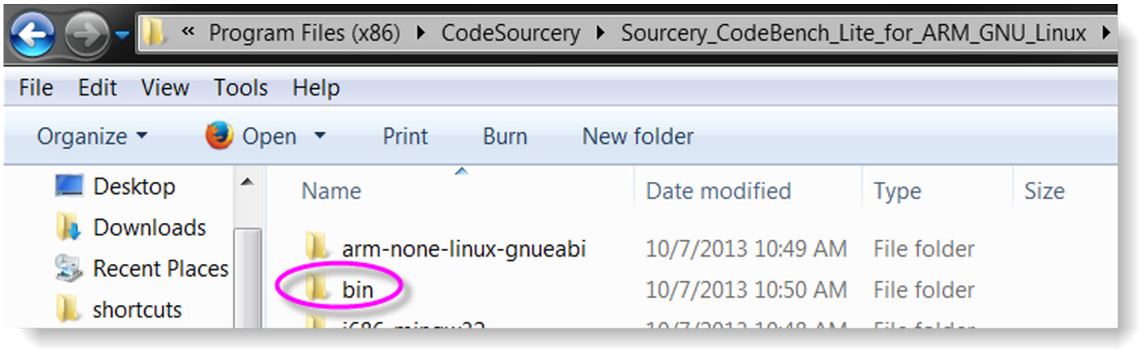
C:\Program Files (x86)\CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_GNU_Linux

Press "Done" to quit the installer.

12. Use copy and paste in a Windows browser to navigate to this location:

```
C:\Program Files (x86)\CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_GNU_Linux
```

13. You will see several folders inside of the Sourcery_CodeBench_Lite_for_ARM_GNU_Linux folder. One of the folders is named bin. ("Bin" means "binaries".)



Name	Date modified	Type	Size
arm-none-linux-gnueabi	10/7/2013 10:49 AM	File folder	
bin	10/7/2013 10:50 AM	File folder	
i686-mingw32	10/7/2013 10:48 AM	File folder	

Note the location of the bin folder, as you will need this path later. (See Set The Paths (Pg. 33). Example location (same as path for CodeBench Lite step 12. above):
C:\Program Files (x86)\CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_GNU_Linux:

Note For Using Sourcery CodeBench Lite With Windows 8 Operating System
As of the publication date of the guide, Mentor Graphics does not support Sourcery CodeBench Lite on the Windows 8 platform. However, we found a workaround. If you chose the defaults during the installation of Sourcery CodeBench Lite, the application binaries will have been installed under the current user.

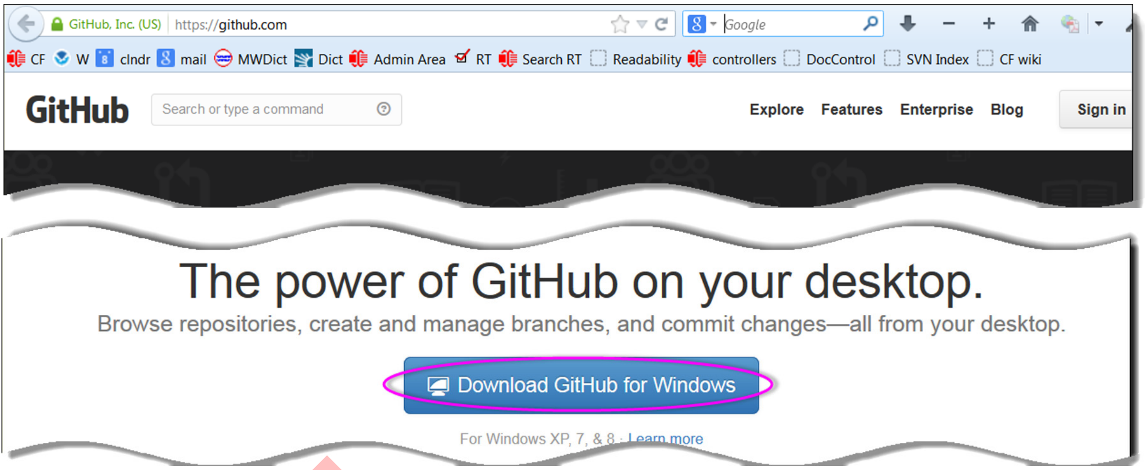
1. Go to the directory via a command prompt:
   `CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_GNU_Linux\bin` and type: `set PATH=%PATH%;%CD%`
   This will add the current directory to your path.
2. Unlike Windows 7, the tools cannot be run from any directory. You must create a special directory from which the tools will run. Create a working directory (in this example we will call it source) under:
   `CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_GNU_Linux`
3. Use the `cd` command to change/create the new directory and you are ready to build applications.

GIT

External libraries are not required to program the CFA10036. There are free sample code files with documentation available on GitHub. GitHub is an open source revision control system for software development projects. The program *Git* is required to download the CFA10036 files from the GitHub repository github.com/crystalfontz/cfa_10036_kernel).

1. Go to https://github.com.

2. Scroll down to find the blue button *Download GitHub for Windows*.

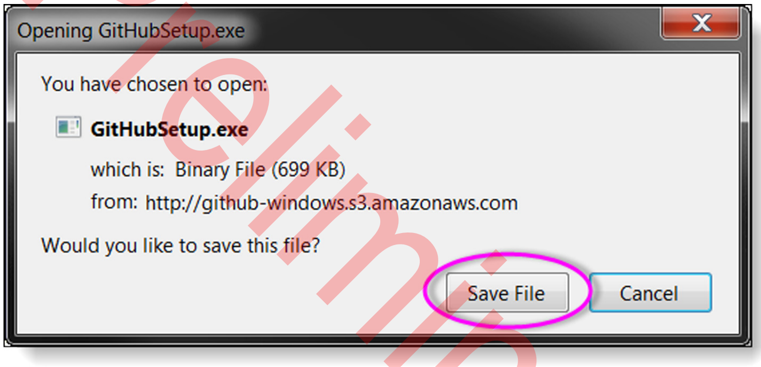


The power of GitHub on your desktop.

Browse repositories, create and manage branches, and commit changes—all from your desktop.

Download GitHub for Windows
For Windows XP, 7, & 8 · Learn more

3. Click on the blue button “Download GitHub for Windows”. You will be prompted to save *GitHubSetup.exe*. Save the file to your computer.


Opening GitHubSetup.exe

You have chosen to open:

GitHubSetup.exe

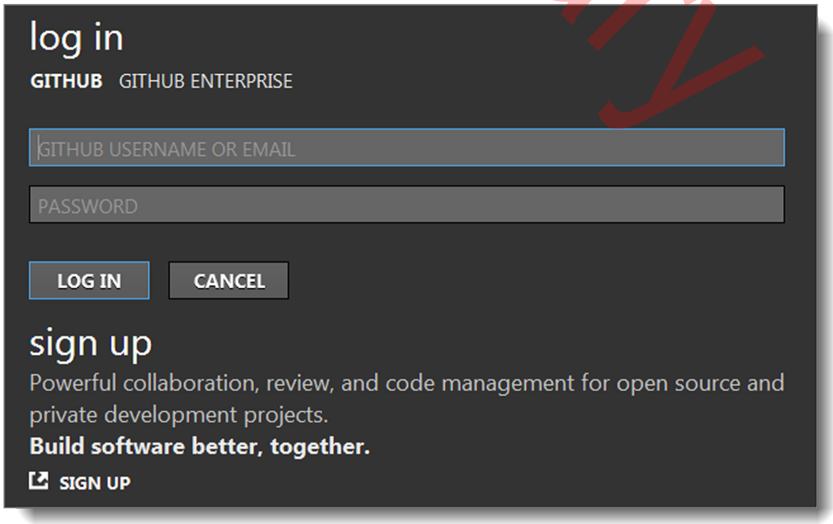
which is: Binary File (699 KB)

from: http://github-windows.s3.amazonaws.com

Would you like to save this file?

Note the location where you saved the file.

4. Go the location where you downloaded *GitHubSetup.exe* file and double click on it. After a few moments, this screen will display:


log in

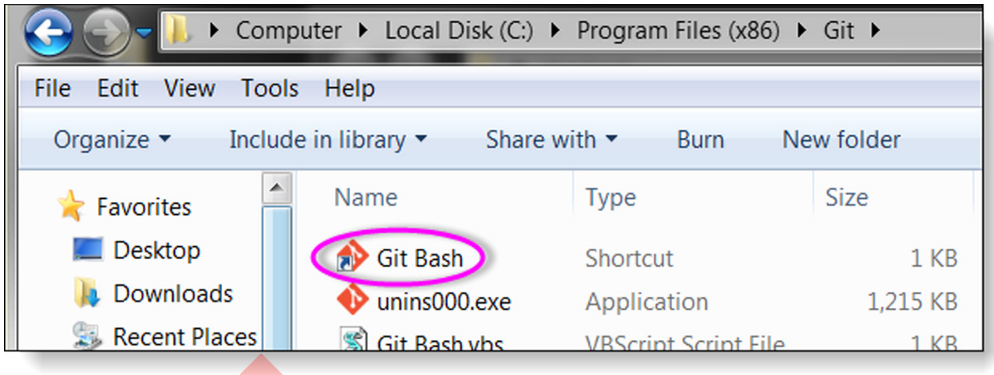
sign up

Powerful collaboration, review, and code management for open source and private development projects.

Build software better, together.

5. If you do not have a GitHub account, click on the “SignUp” button. The GitHub web page with the heading *Plans & Pricing* will display. Choose a plan and follow the instructions to create an account.

6. After you sign in, follow the *Git Setup Wizard* instructions. The default options are fine. The Git program will be saved in your computer's *Program Files (x86)* folder. Example location: `C:\Program Files (x86)\Git`.
7. Find the *Git Bash* shortcut and make a copy of the shortcut in a convenient location. You will need Git Bash to open a terminal window for programming later.

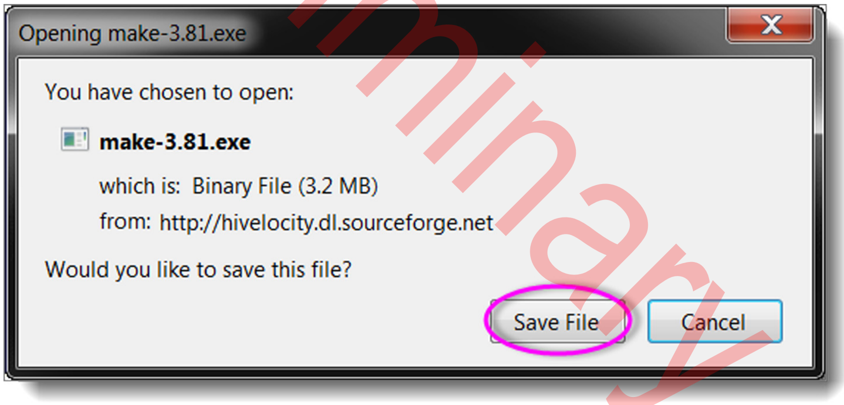

Computer ▸ Local Disk (C:) ▸ Program Files (x86) ▸ Git ▸
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ Burn New folder
Favorites
Desktop
Downloads
Recent Places

Name	Type	Size
Git Bash	Shortcut	1 KB
unins000.exe	Application	1,215 KB
Git Bash.vbs	VBScript Script File	1 KB

GNU MAKE

GNU Make is a key part of the build systems used for development. Make is part of the free GnuWin download. No registration is required.

1. Go to the SourceForge web page gnuwin32.sourceforge.net/downlinks/make.php. You will be prompted to download an .exe file (*make-VERSION.exe*). Save the file to your computer.



You have chosen to open:

make-3.81.exe

which is: Binary File (3.2 MB)

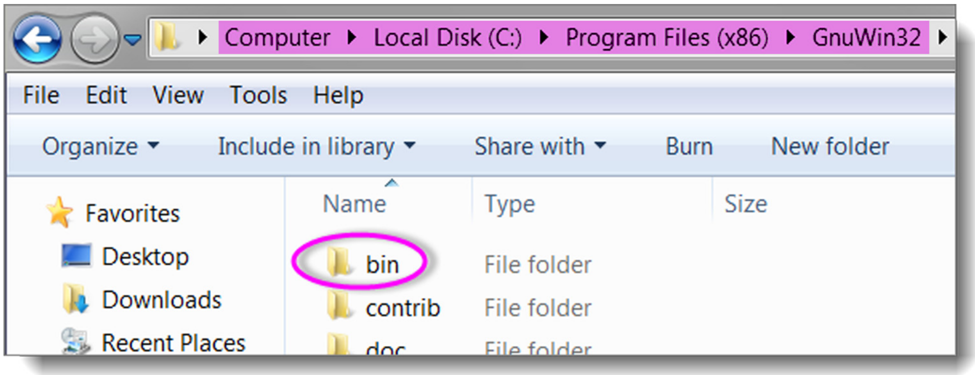
from: http://hivelocity.dl.sourceforge.net

Would you like to save this file?

Note the location where you saved the file.

2. Go to the location where you downloaded the *make-VERSION.exe* file and double click on it. The installation wizard *Make Setup Wizard* will open. Follow the *Make Setup Wizard* instructions. The GnuWin32 program will be saved in your computer's Program Files (x86) folder.

3. Note the installation path of the bin folder within the GnuWin32 program files folder, as you will need this path later. (See section Set The Paths (Pg. 33).)
Example of location:


Computer ▸ Local Disk (C:) ▸ Program Files (x86) ▸ GnuWin32 ▸
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ Burn New folder
Favorites
Desktop
Downloads
Recent Places
Name
Type
Size
bin
File folder
contrib
File folder
doc
File folder

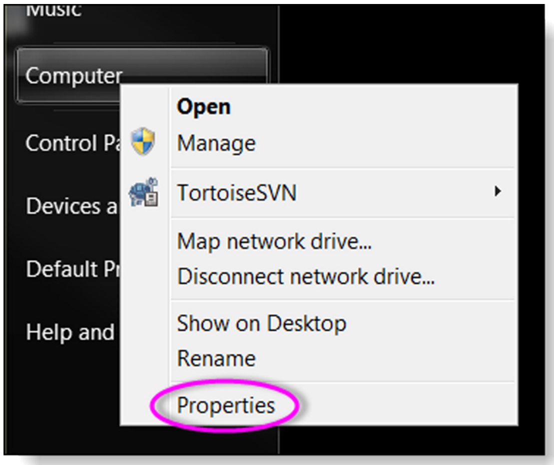
1. For convenience, you might want to right click in the File Explorer toolbar (previously called Window's Explorer).
2. In the dropdown menu, select "copy address as text".
3. Paste the path in Window's Notepad (.txt file) and save for reference. The path will look something like this:
`C:\Program Files (x86)\GnuWin32`

SET THE PATHS

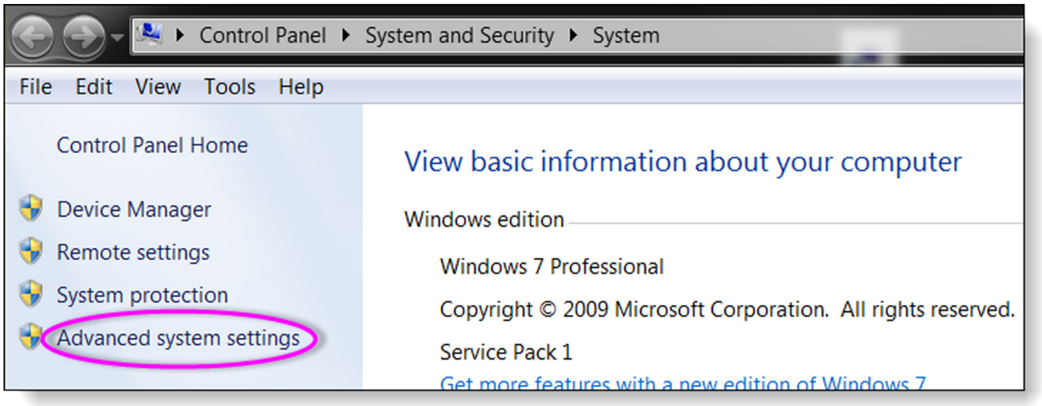
Setting the paths will ensure that the compiler and linker will know which folders to look in for the libraries and their associated files. On your computer, setting the paths may not be necessary. If you want to see if you can skip this step, move on to Sublime Text (Pg. 37) and complete the remaining instructions. If an error returns when you run Build, you to return to this section and follow the instructions to set the paths.

To set the paths, you need to add (1) path for *Sourcery CodeBench Lite bin folder* (i.e., cross compiler's binaries directory) and (2) path for *GnuWin32 bin folder* (i.e., make directory) to your computer's system's path variable. This requires opening several windows, described in the instructions below in this order: *Computer>Properties>Advanced System Settings>System Properties>Environment Variables>Edit System Variable*.

1. Click on your PC's "Start" menu icon.
2. Find "Computer" and right-click on it to display a menu.
3. On the menu, click on "Properties" to open the *System* window.



4. On *System* window, click to select “Advanced system settings” window. The *System Properties* window opens.



View basic information about your computer

Windows edition

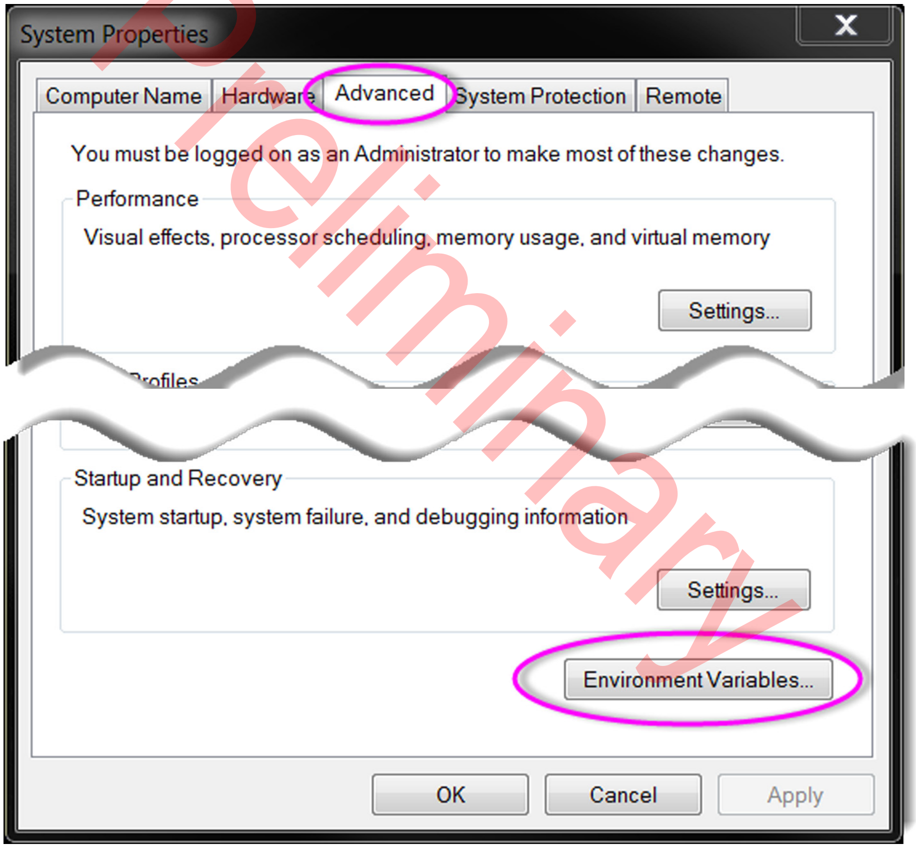
Windows 7 Professional

Copyright © 2009 Microsoft Corporation. All rights reserved.

Service Pack 1

Get more features with a new edition of Windows 7

5. In *System Properties* window, click on the *Advanced* tab. On the *Advanced* tab, click on the *Environment Variables...* button. *Environment Variables* window will open.



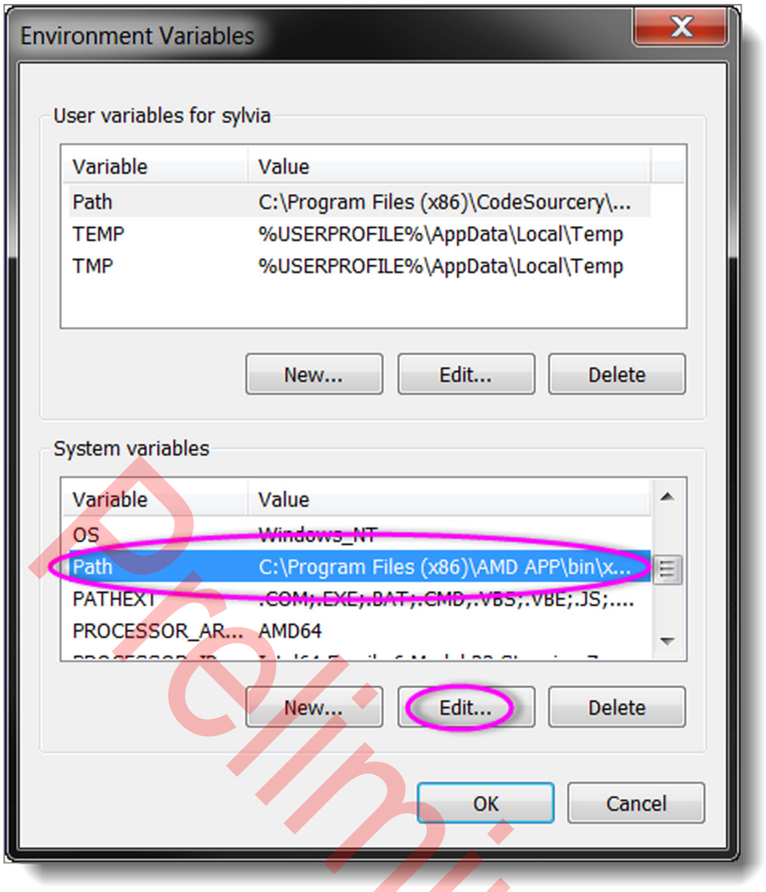
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

System startup, system failure, and debugging information

6. The *Environment Variables* window has two boxes: *User variable* and *System variable*.
   1. In the *System variables* box, scroll to find *Path* in the *Variable* column.
   2. Click on *Path* to highlight it.

3. Click on the “Edit” button.



Environment Variables

User variables for sylvia

Variable	Value
Path	C:\Program Files (x86)\CodeSourcery\...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

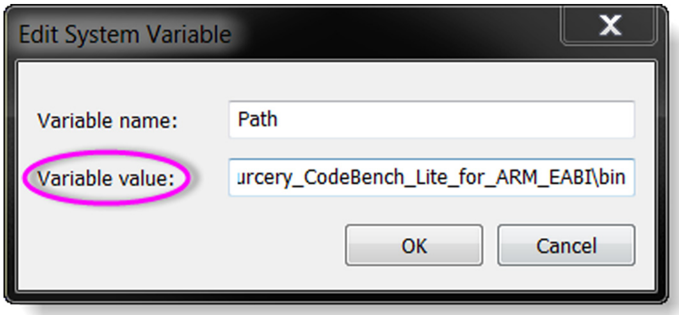
Variable	Value
OS	Windows_NT
Path	C:\Program Files (x86)\AMD APP\bin\x...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_AR...	AMD64

The *Edit System Variable* window will open.

4. The *Edit System Variable* window has two text boxes: *Variable name* and *Variable value*. Depending what you already have installed on your PC, you may have a long path consisting of elements installed by different programs. Unfortunately, window cannot be enlarged to see the entire path at one time. Use your keyboard’s right arrow key to scroll through the entire path. Note that there are semicolons (`;`) between the elements of the path.

   In the instructions below, you will first add the path for the *Sourcery CodeBench Lite bin* folder and then add the path for the *GnuWin32 bin* folder.

   Tip: You might want to prepare this path in Windows Notepad, then copy and paste the path into the Variable value field.



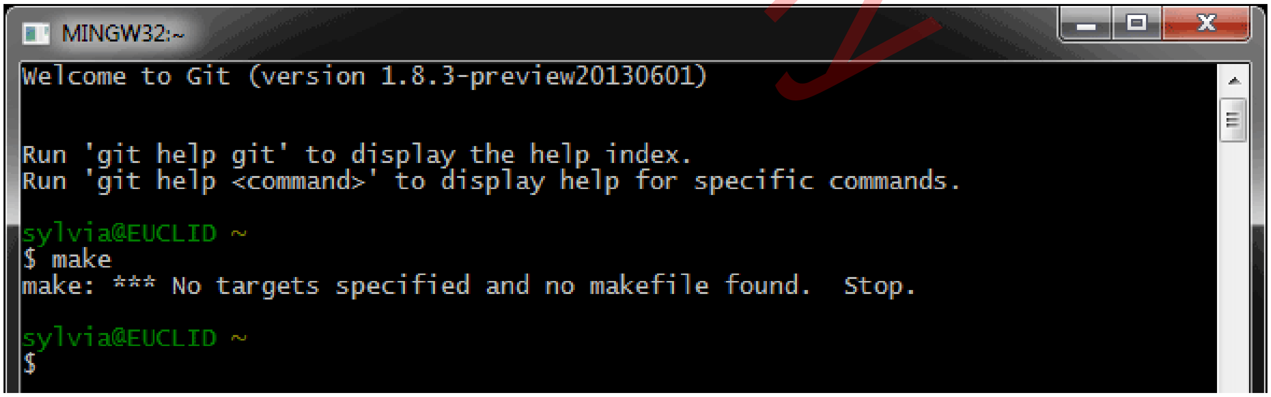
Edit System Variable

Variable name: Path

Variable value: urcery_CodeBench_Lite_for_ARM_EABI\bin

Before adding each of the two paths, you will need to type in a semicolon (`;`) at the end of the path already listed. Here are the steps:

1. If there is no semicolon at the END of the text in the *Variable value* box, add one: `;`. Do not click on the "OK" button.
2. Go to the bin folder within the *Sourcery_CodeBench_Lite_for_ARM_GNU_Linux* you installed earlier. (See Sourcery CodeBench Lite bin folder). The path will look something like this:
   `C:\Program Files (x86)\CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_GNU_Linux`
3. Copy the bin folder path from the File Explorer (previously called Window's Explorer) address bar.
4. Paste this path after the semicolon at the END of the text in the *Variable value* box.
5. After your paste the bin folder path, add another semicolon: `;` . Do not click on the "OK" button.
6. Go to the bin folder within the WinGnu32 you installed earlier. (See GNU32 bin folder.) The path will look something like this: `C:\Program Files\GnuWin32\bin`
7. Copy the bin folder path from the File Explorer address bar. Paste this after the semicolon at the end of the text.
8. In addition to what the path already listed before you added these paths, the *end* of the path will now look something like this:
   `;C:\Program Files\CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_GNU_Linux;C:\Program Files (x86)\GnuWin32`
9. Click on the "OK" buttons for the series of windows that you opened.
10. Open a terminal by clicking on the *Git Bash* shortcut. Confirm that these steps were successful by typing `make` in the terminal. The result should look like this:



MINGW32:~

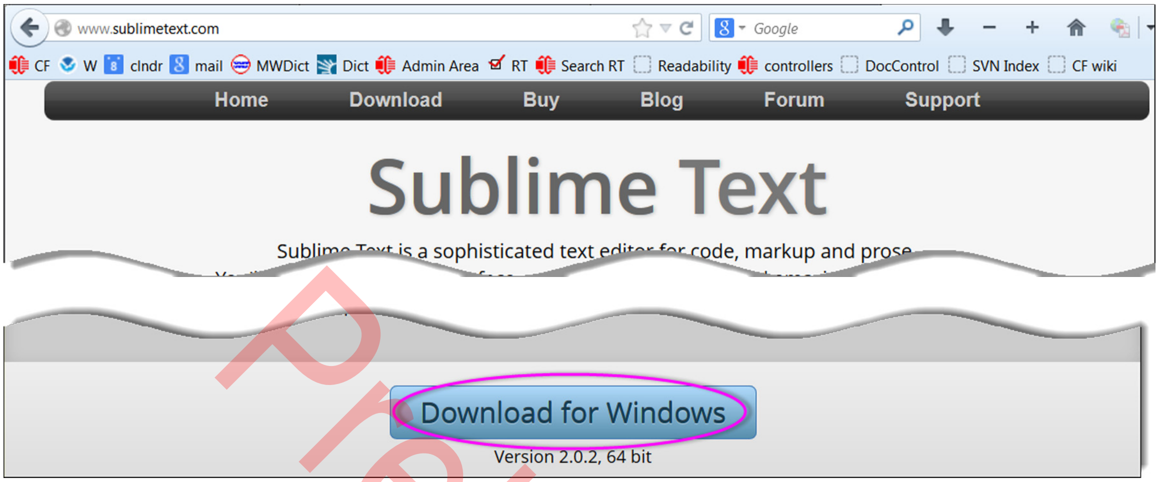
```
Welcome to Git (version 1.8.3-preview20130601)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

sylvia@EUCLID ~
$ make
make: *** No targets specified and no makefile found.  Stop.

sylvia@EUCLID ~
$
```

The message "*No targets specified and no makefile found*" is good. This means that make is working. For now, close this window and continue with the instructions below to install Sublime Text.

SUBLIME TEXT

To write our Hello World program described below (see WINDOWS: PHYSICALLY CONNECT THE CFA10036 TO YOUR PC (Pg. 42), we used the text editor *Sublime Text 2.* As of 2013-10-22, a beta version of Sublime Text 3 is also available. We we could have used Windows Notepad for the text editing, but Notepad does not have the build and make tool that are needed after the code is written.

1. Go to www.sublimetext.com. The suggested download for your operating system will be shown as a blue button.



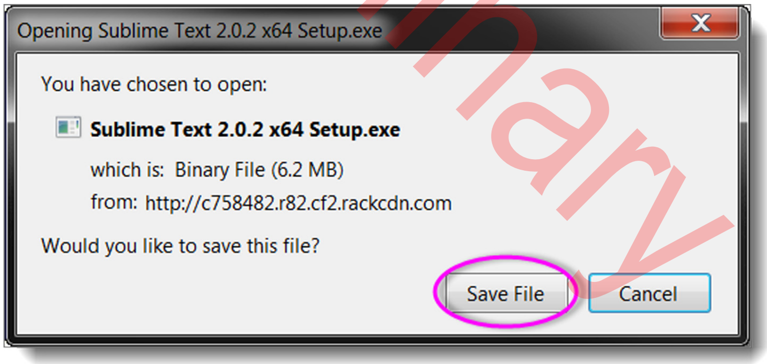
Sublime Text is a sophisticated text editor for code, markup and prose

Version 2.0.2, 64 bit

When your mouse arrow hovers over the button, a popup with links to other platforms displays.

This free program has no enforced time limit, but occasionally has a popup display that requests that you purchase the full version (~$70 in October 2013).

2. Click on the “Download for Windows” button. You will be prompted to save an .exe file (*Sublime Text VERSION Setup.exe*). Save the file to your computer.



Opening Sublime Text 2.0.2 x64 Setup.exe

You have chosen to open:

Sublime Text 2.0.2 x64 Setup.exe

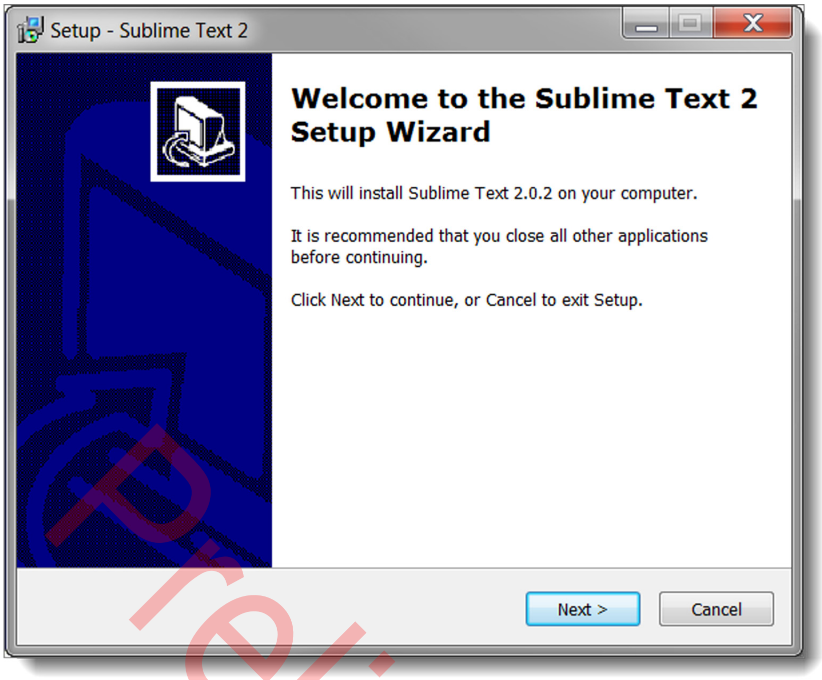
which is: Binary File (6.2 MB)

from: http://c758482.r82.cf2.rackcdn.com

Would you like to save this file?

Note the location where you saved the file.

3. Go to the *Sublime Text VERSION Setup.exe* and double click on it. The installation wizard *Sublime Text Setup Wizard* will display.



Welcome to the Sublime Text 2 Setup Wizard

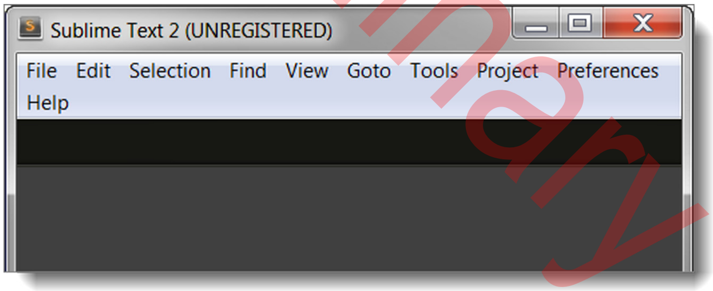
This will install Sublime Text 2.0.2 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Note the location of the installation. For convenience, you might want to checkmark *Add to explorer context menu* during installation.

4. Go to the location where you downloaded the *Sublime Text* program. Find *sublime_text.exe* and create a *Sublime Text* shortcut in a convenient location. If you click on the *Sublime Text* shortcut, a window will open that looks similar to a Windows terminal window.



For now, close this window and continue with the instructions to install WinSCP.

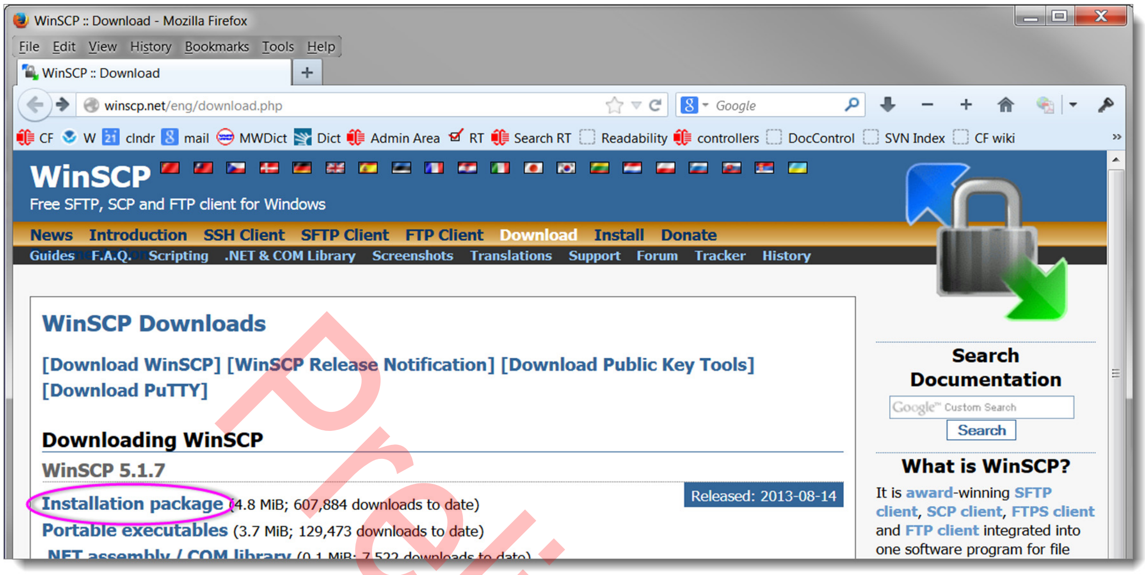
WINSCP

An SSH File Transfer Protocol (SFTP) is needed for your Windows operating system to communicate with a Linux device, such as the CFA10036.

For these instructions, we used WinSCP (Windows Secure Copy), a free open-source SFTP/SCP/FTP client for Windows. The program uses SCP to transfer binary files from your PC to the CFA10036. The interface is simple, with drag and drop functionality.

WinSCP can also be used as a remote editor. When you click on a (text) file in your PC's remote file manager, it transfers the file to and from the PC and CFA10036 and opens the file in an integrated editor. The integrated editor gives you direct access to the code.

1. Go to http://winscp.net/eng/download.php.
2. The web page *WinSCP Downloads* will display.
3. Look for the *Installation package* of the most recent version of WinSCP. We used WinSCP 5.1.7 release 2013-08-14.


WinSCP :: Download - Mozilla Firefox
File Edit View History Bookmarks Tools Help
WinSCP :: Download
winscp.net/eng/download.php
Google
CF W cIndr mail MWDict Dict Admin Area RT Search RT Readability controllers DocControl SVN Index CF wiki
WinSCP
Free SFTP, SCP and FTP client for Windows
News Introduction SSH Client SFTP Client FTP Client Download Install Donate
Guides F.A.Q. Scripting .NET & COM Library Screenshots Translations Support Forum Tracker History
Search Documentation
Google Custom Search
Search

WinSCP Downloads

[Download WinSCP] [WinSCP Release Notification] [Download Public Key Tools] [Download PuTTY]

Downloading WinSCP

WinSCP 5.1.7

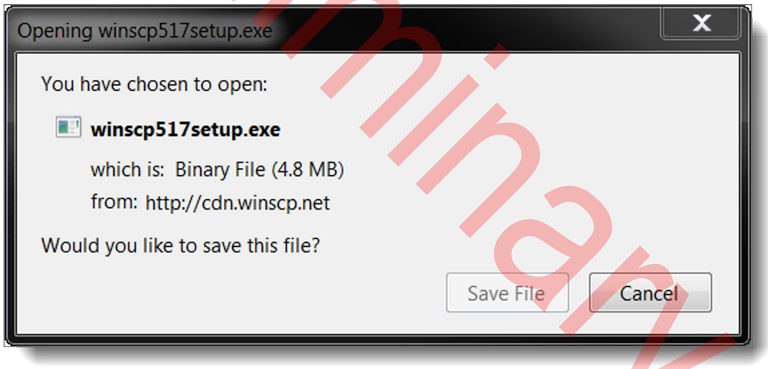
Released: 2013-08-14

Installation package (4.8 MiB; 607,884 downloads to date)
Portable executables (3.7 MiB; 129,473 downloads to date)
.NET assembly / COM library (0.1 MiB; 7,522 downloads to date)

What is WinSCP?

It is award-winning SFTP client, SCP client, FTPS client and FTP client integrated into one software program for file

4. You will be prompted to save an exe file (*winscpVERSIONsetup.exe)*. Save the file to your computer.



You have chosen to open:

winscp517setup.exe

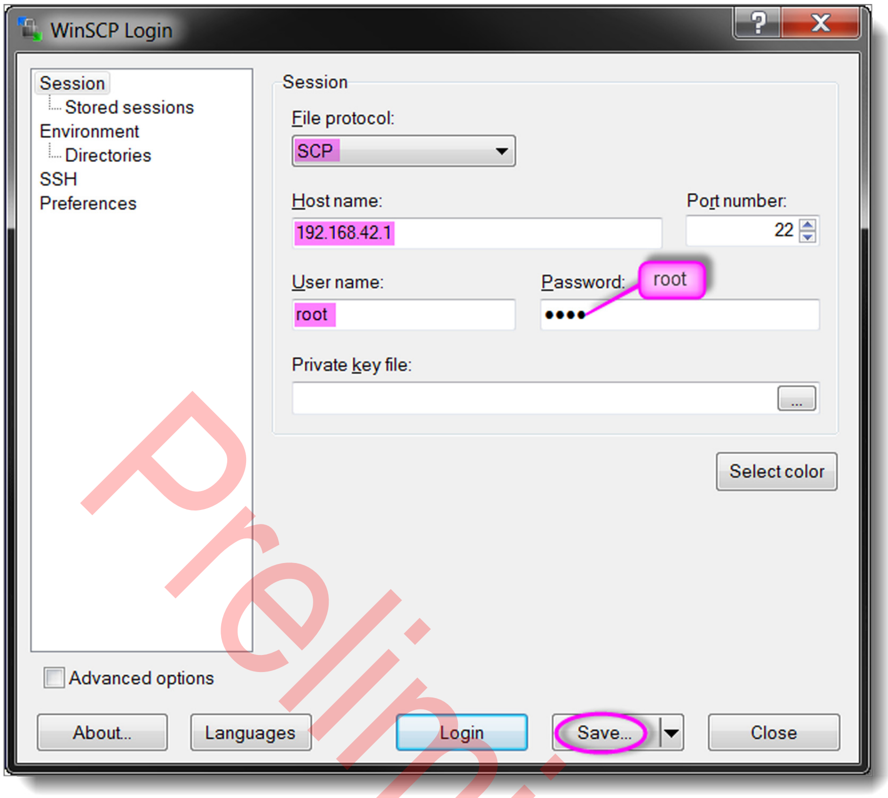
which is: Binary File (4.8 MB)
from: http://cdn.winscp.net

Would you like to save this file?

Note the location where you saved the file.

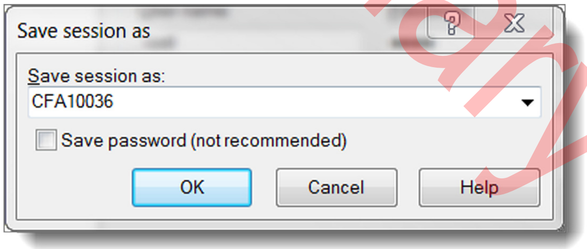
5. Go to the *winscpVERSIONsetup.exe* file and double click on it. The installation wizard *WinSCP Setup Wizar*d will display.
6. Follow the *WinSCP Setup Wizard* instructions. The default options are fine. The WinSCP program will be saved in your computer's *Program Files (x86)* folder. Example location: `C:\Program Files (x86)\WinSCP`.
7. Find *WinSCP.exe* and create a *WinSCP* shortcut in a convenient location.
8. Click on the *WinSCP* shortcut. The *WinSCP Login* window will open. Use this information to complete the fields:
   File protocol: select *SCP* from dropdown menu.
   Host name: type `192.168.42.1`
   Port number: Default is `22`.

User name: `root`
Password: `root`
Private key file: Leave blank.



File protocol: SCP
Host name: 192.168.42.1
Port number: 22
User name: root
Password: •••• root
Private key file:

9. Click on the “Save” button. A window with will open with the heading *Save session as.*
10. You can keep the default session name (IP address) or type a different name. For these instructions, chose the session name *CFA10036*.



Save session as: CFA10036
□ Save password (not recommended)

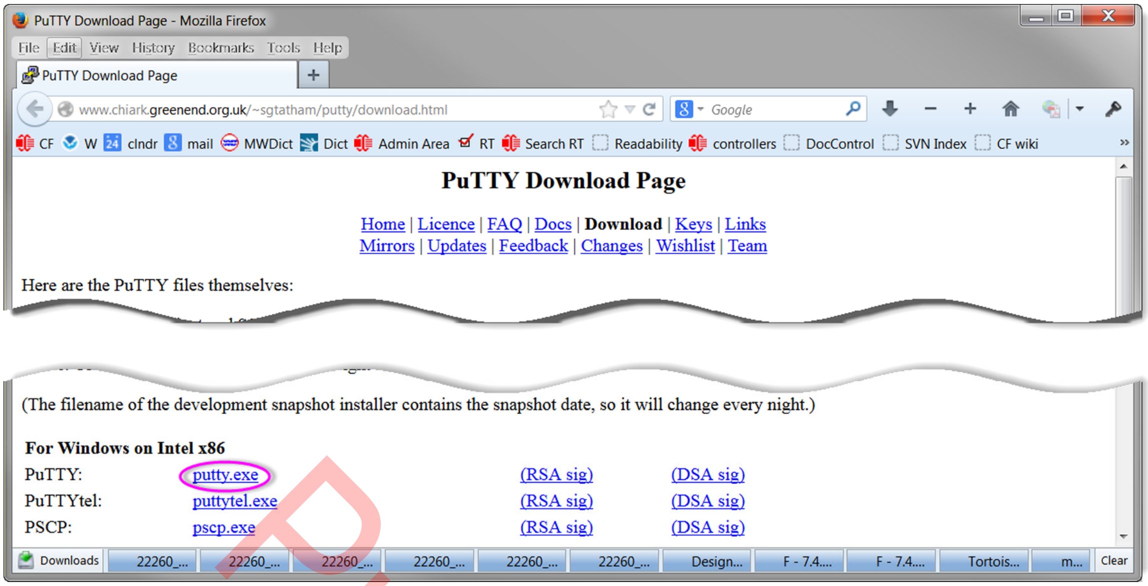
11. Click on the “OK” button after you name the session.
12. The *WinSCP Login* window reopens. For now, close this window and continue with the instructions below to install PuTTY.

PUTTY

PuTTY is an SSH and telnet client. You can use PuTTY to open a terminal to the CFA10036 device and run your program from there.

1. Go to URL www.chiark.greenend.org.uk/~sgtatham/putty/download.html.
2. Scroll down the web page and find the section heading *Binaries*. Under this heading, find the subheading *For Windows on Intel 86.* You may have a choice between a current release or a beta release. For these instructions, we chose the current release.

3. Click on the *putty.exe* link.


PuTTY Download Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
PuTTY Download Page
www.chiark.greenend.org.uk/~sgtatham/putty/download.html
Google
CF W cldr mail MWDict Dict Admin Area RT Search RT Readability controllers DocControl SVN Index CF wiki
Downloads 22260_... 22260_... 22260_... 22260_... 22260_... 22260_... Design... F - 7.4.... F - 7.4.... Tortois... m... Clear

PuTTY Download Page

Home | Licence | FAQ | Docs | Download | Keys | Links
Mirrors | Updates | Feedback | Changes | Wishlist | Team

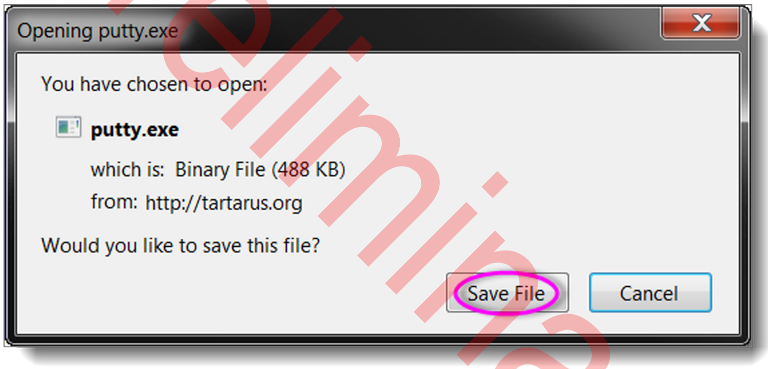
Here are the PuTTY files themselves:

(The filename of the development snapshot installer contains the snapshot date, so it will change every night.)

For Windows on Intel x86

PuTTY:	putty.exe	(RSA sig)	(DSA sig)
PuTTYtel:	puttytel.exe	(RSA sig)	(DSA sig)
PSCP:	pscp.exe	(RSA sig)	(DSA sig)

4. You will be prompted to save *putty.exe* file. Save this small file to your computer.



You have chosen to open:

putty.exe

which is: Binary File (488 KB)
from: http://tartarus.org

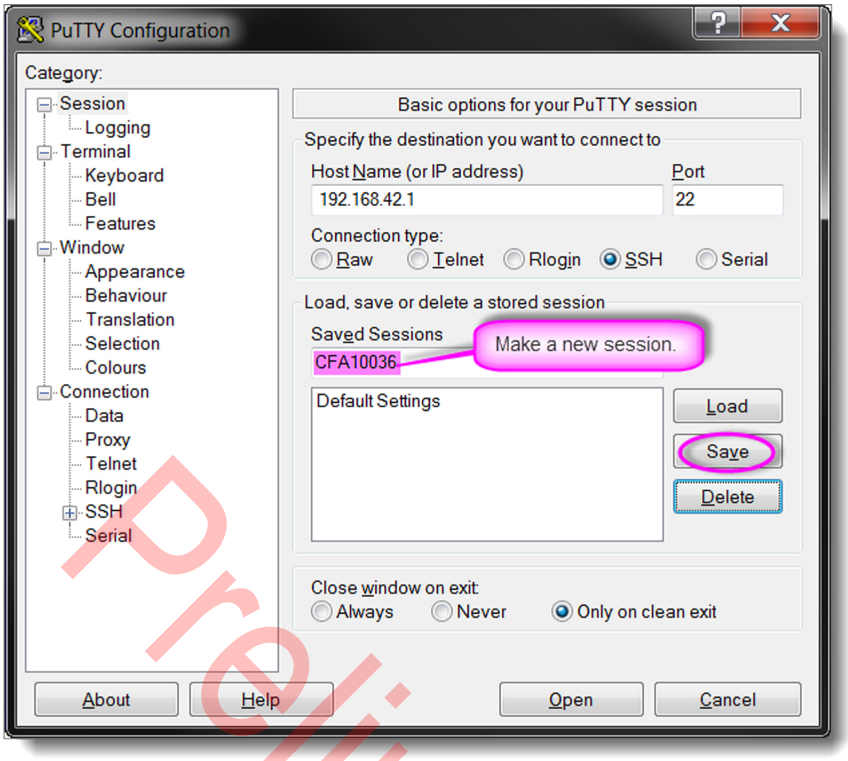
Would you like to save this file?

Note the location where you saved the file.

5. Find the *putty.exe* file and create a shortcut in a convenient location.
6. Click on the *PuTTY* shortcut. The *PuTTY Configuration* window will open. You will now make and save a "session" that you can reuse when communicating between the CFA10036 and your PC.

   Use this information to complete the fields:
   Host name: type `192.168.42.1`
   Port: Default is `22`. This is OK.
   Connection type: Default is `SSH`. This is OK.
   Saved Sessions: Type `CFA10036`.

7. Click on the “Save” button.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.42.1

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: CFA10036

Default Settings

Make a new session.

Close window on exit: ○ Always ○ Never ● Only on clean exit

Note that CFA10036 session is added to the *Saved Sessions* list below *Default Settings*.

8. For now, close the PuTTY configuration window and continue with the instructions below to physically connect the CFA10036 to your PC.

WINDOWS: PHYSICALLY CONNECT THE CFA10036 TO YOUR PC

Crystalfontz sells the CFA10036 as an assembled product with additional components. See ADD-ON PRODUCTS FOR THE CFA10036 (Pg. 20).

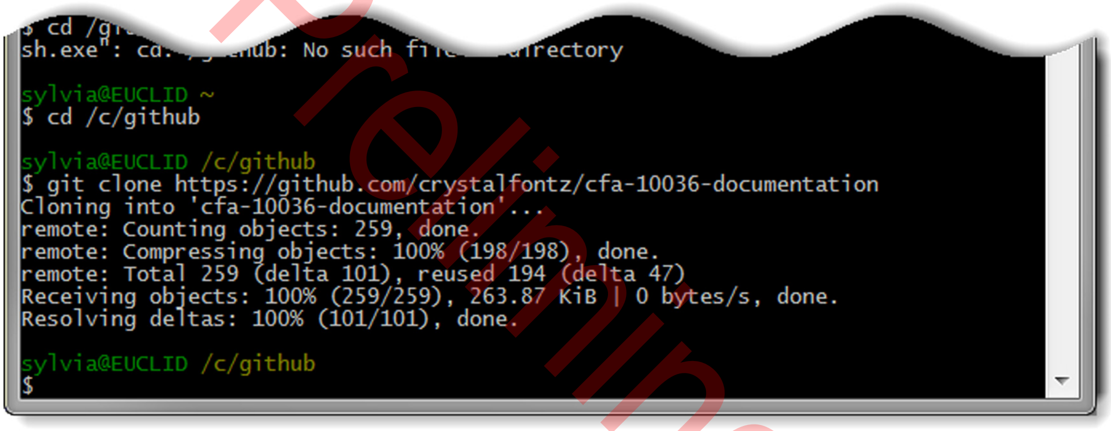
For the remainder of the instructions in this *Software Quick Start Guide*, we chose to use the kit CFA100366. This kit includes a small preinstalled OLED, the CFAL12832D-B.

1. Before you connect the CFA10036 to your PC, confirm that the SD card shipped with the CFA10036 is inserted in the CFA10036’s card reader.
2. For a USB connection, use the Crystalfontz WR-USB-Y27 cable. The cable has two different types of USB connectors, one smaller than the other. Connect the cable’s smaller micro-B USB female connector to the CFA10036’s micro-B USB connector. Connect the cable’s larger USB-A female connector to PC’s USB-A connector. When you make the connection, the green “power” LED on the top edge of the CFA10036 will light. At first, the OLED will display static.Then the splash screen with the Crystalfontz name and logo will display.

WINDOWS: DOWNLOAD SAMPLE CODE FROM GITHUB REPOSITORY

1. Create a working directory (folder) to clone (copy) the CFA10036 repository into. We recommend that you create the working directory as close as possible to the root of your PC's drive. For these instructions, we used: `C:\Github`.
2. Click on the *Git Bash* shortcut to open a terminal window. Navigate to the working directory. If using our example, use this command:
   type `cd /c/github`↵
3. To clone the repository, copy and paste this command:
   `git clone https://github.com/crystalfontz/cfa-10036-documentation`
   On some terminals, you may need to use SHIFT+INSERT keys instead of copy and paste.

   Git will automatically download the most recent files and install them into a new folder named *cfa-10036-documentation* in the working (current) directory.



```
$ cd /gi
sh.exe": cd.    hub: No such fi      irectory

sylvia@EUCLID ~
$ cd /c/github

sylvia@EUCLID /c/github
$ git clone https://github.com/crystalfontz/cfa-10036-documentation
Cloning into 'cfa-10036-documentation'...
remote: Counting objects: 259, done.
remote: Compressing objects: 100% (198/198), done.
remote: Total 259 (delta 101), reused 194 (delta 47)
Receiving objects: 100% (259/259), 263.87 KiB | 0 bytes/s, done.
Resolving deltas: 100% (101/101), done.

sylvia@EUCLID /c/github
$
```

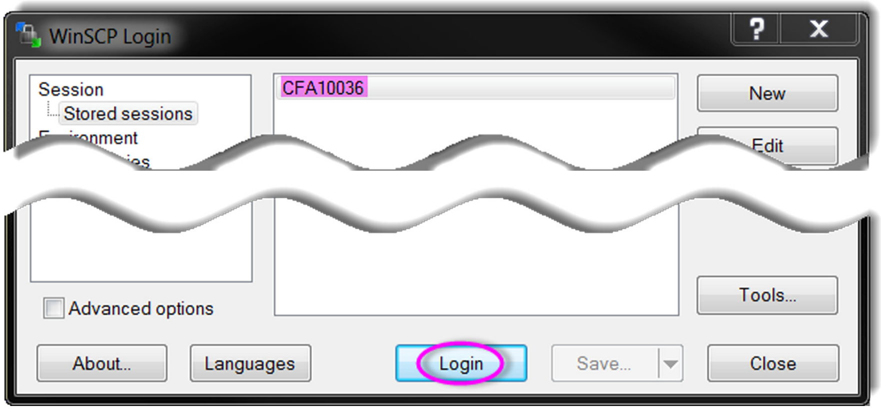
WINDOWS: WRITE AND COMPILE A PROGRAM "HELLO WORLD"

This section describes how to write and compile a simple Linux program "Hello World" for the CFA10036. Although the Linux program can be written and compiled on a computer with a Windows operating system, it cannot be executed. Following this section are instructions on how to execute the program on the CFA10036.

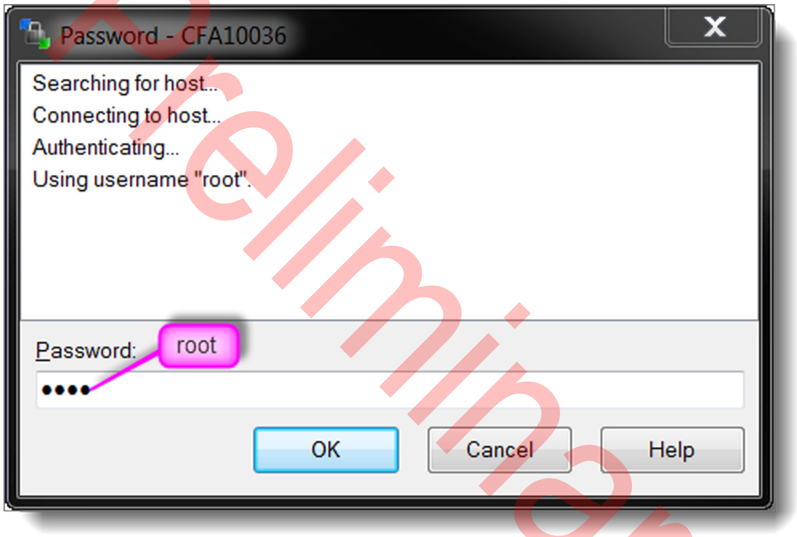
LOGIN TO WINSCP

1. Click on the *WinSCP* shortcut to open the program.
2. Select the *Stored Session CFA10036* that you created earlier.

3. Click on the “Login” button.

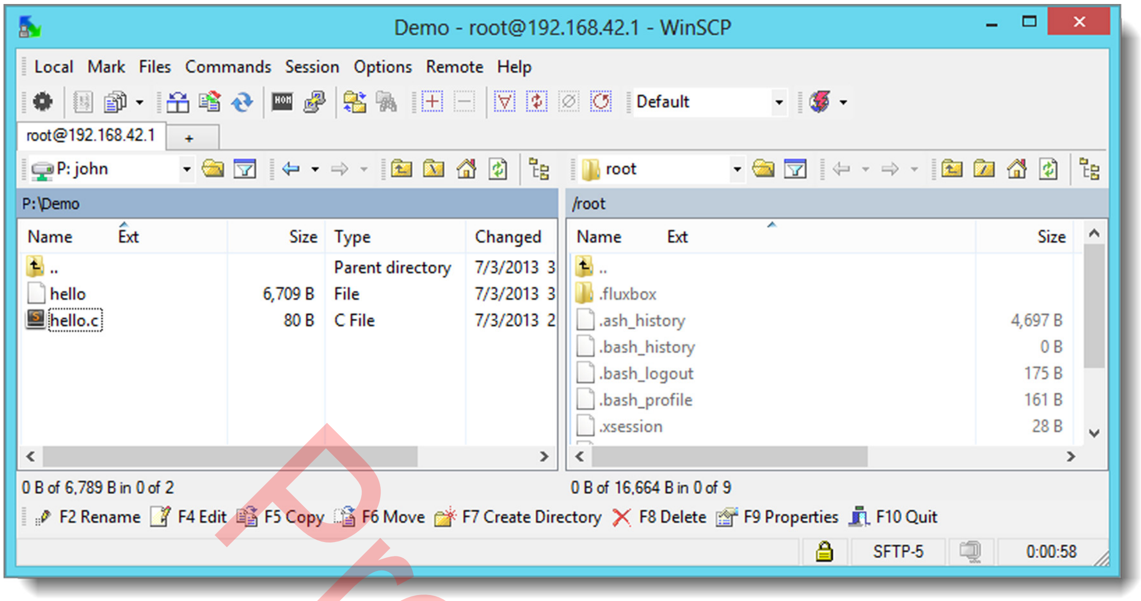


4. The *Password - CFA10036* window opens. You be prompted to enter a password:
Password: type `root`



Searching for host...
Connecting to host...
Authenticating...
Using username "root".

5. The CFA10036 WinSCP window with two panels will open. In the left panel, navigate to your My_Code folder. Drag and drop the *hello* file (not *hello.c*) to transfer it from the left panel to the right panel.


Demo - root@192.168.42.1 - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
root@192.168.42.1
P: john
root
F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit
SFTP-5
0:00:58

P:\Demo

Name	Ext	Size	Type	Changed
..			Parent directory	7/3/2013 3
hello		6,709 B	File	7/3/2013 3
hello.c		80 B	C File	7/3/2013 2

0 B of 6,789 B in 0 of 2

/root

Name	Ext	Size
..		
.fluxbox		
.ash_history		4,697 B
.bash_history		0 B
.bash_logout		175 B
.bash_profile		161 B
.xsession		28 B

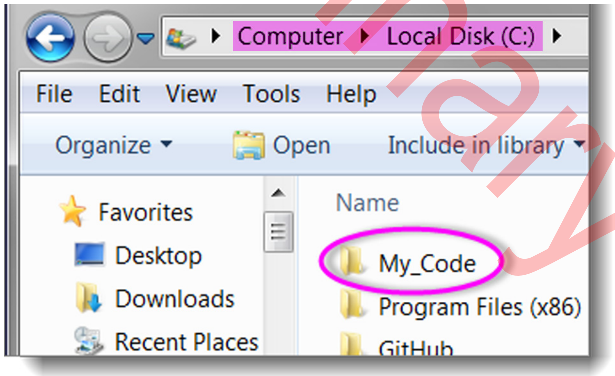
0 B of 16,664 B in 0 of 9

After the transfer is finished, close the WinSCP window

WRITE “HELLO WORLD” CODE USING SUBLIME TEXT

This section describes how to write a simple “Hello World” program in C.Reminder to me: .

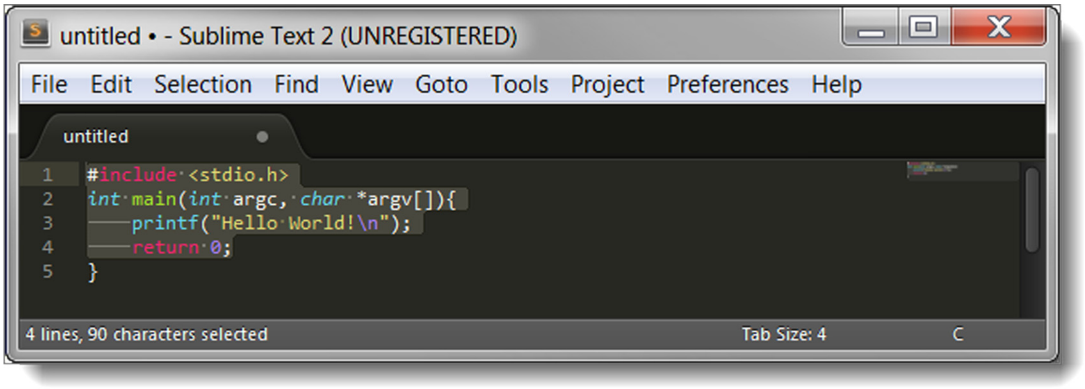
1. Create a folder close to your working directory to hold the files for your program. We chose to name our folder *My_Code*.



2. Open *Sublime Text* and select “File>New” to create a new empty window.
3. Copy and paste the text below into the Sublime Text window:

```
#include <stdio.h>
int main(int argc, char *argv[])
   {
 printf("Hello World!\n");
 return 0;
   }
```

4. Colors will be automatically applied to the text.



```
#include <stdio.h>
int main(int argc, char *argv[]){
	printf("Hello World!\n");
	return 0;
}
```

4 lines, 90 characters selected

Tab Size: 4

C

5. Save this Sublime Text file in your working directory as *hello.c*. The .c extension represents C code.

COMPILE “HELLO WORLD” CODE USING GNU MAKE

Now that you’ve written the hello world program, it needs to be compiled using *GNU make*. Make will compile the C code file into objects (.o extensions) and link them all together with the required libraries at the end.

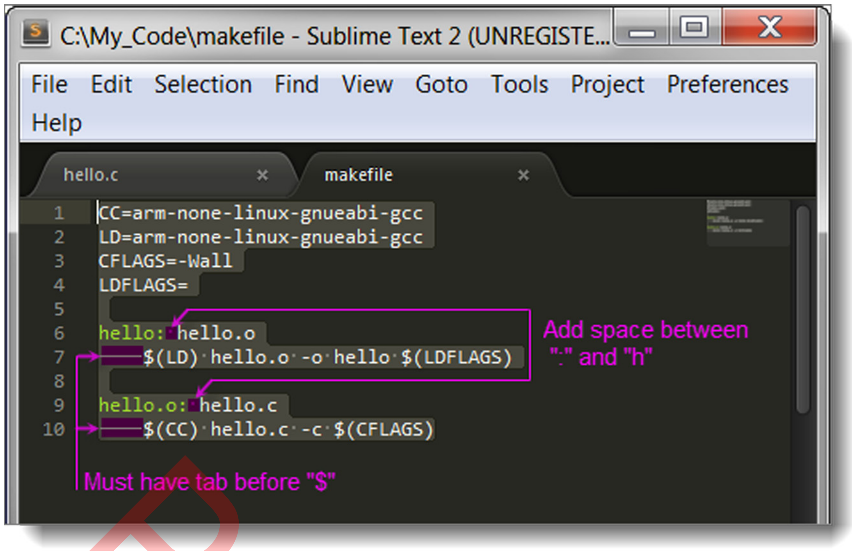
To understand this process, we suggest that you look at the GNU make documentation and examples here: www.gnu.org/software/make/manual/make.html.

6. Create a new Sublime Text file and save it with the file name *makefile* with no extension. Save this file in the same folder as the *hello.c* file.
7. Copy & paste this text into the new makefile:

```
CC=arm-none-linux-gnueabi-gcc
LD=arm-none-linux-gnueabi-gcc
CFLAGS=-Wall
LDFLAGS=

hello: hello.o
	$(LD) hello.o -o hello $(LDFLAGS)

hello.o: hello.c
	$(CC) hello.c -c $(CFLAGS)
```

Note

In the Hello World code highlighted in gray above, two lines of the text must be indented by using the Tab key. See the indented text in the screen capture below this note.

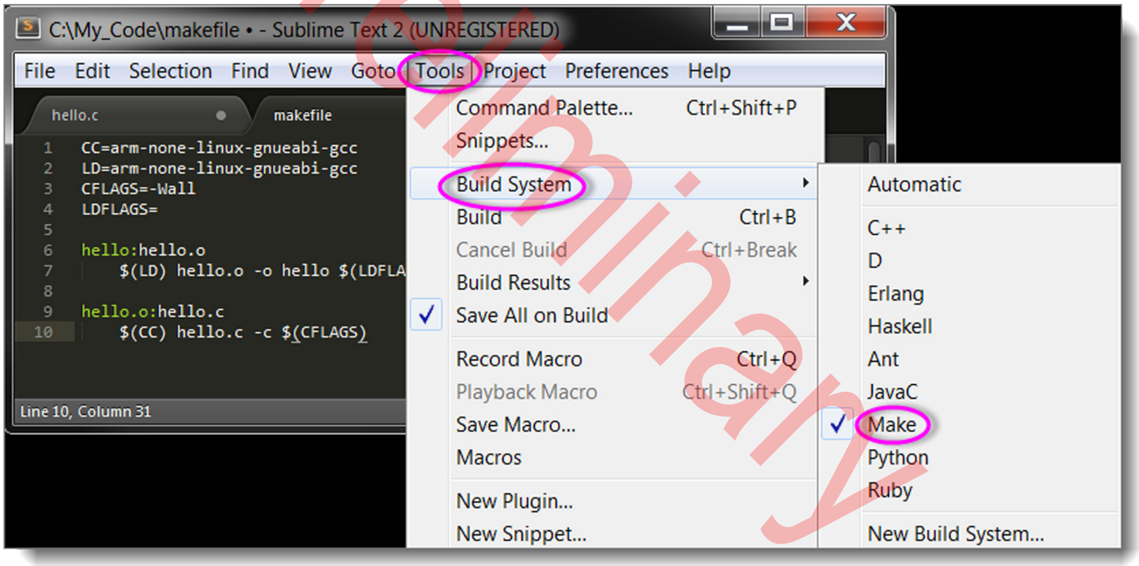

C:\My_Code\makefile - Sublime Text 2 (UNREGISTE...
File Edit Selection Find View Goto Tools Project Preferences Help
hello.c
makefile

```
CC=arm-none-linux-gnueabi-gcc
LD=arm-none-linux-gnueabi-gcc
CFLAGS=-Wall
LDFLAGS=

hello: hello.o
	$(LD) hello.o -o hello $(LDFLAGS)

hello.o: hello.c
	$(CC) hello.c -c $(CFLAGS)
```

Add space between ":" and "h"

Must have tab before "$"

8. Now the Build System needs to be set to Make. In the open Sublime Text makefile, select “Tools>Build System>Make”, as shown here:



```
CC=arm-none-linux-gnueabi-gcc
LD=arm-none-linux-gnueabi-gcc
CFLAGS=-Wall
LDFLAGS=

hello:hello.o
    $(LD) hello.o -o hello $(LDFLA

hello.o:hello.c
    $(CC) hello.c -c $(CFLAGS)
```

Line 10, Column 31

This confirms that Make is checkmarked. The content of windows does not change.

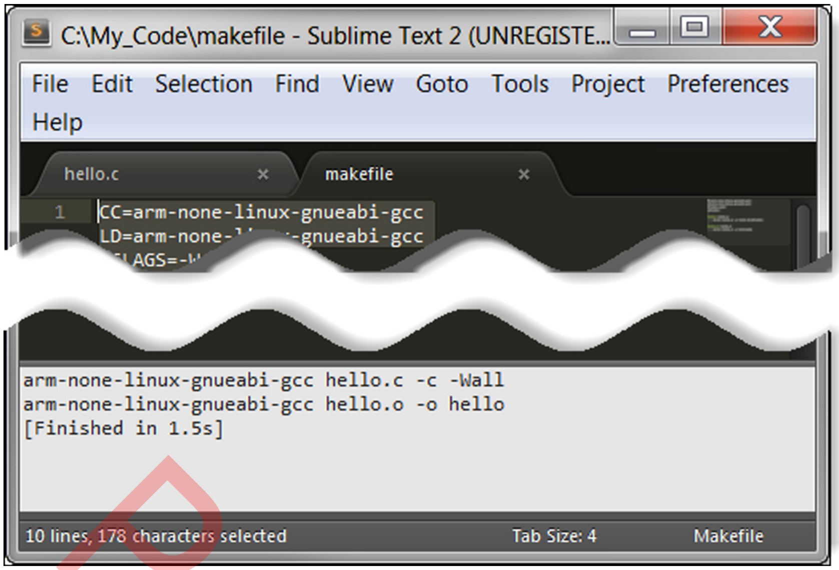
9. The program is ready to build.

Three Ways To Build

You will be building frequently as you develop programs for the CFA10036. Use any of these methods to build:

1. With *makefile* open, select “Tools>Build”.
2. With *makefile* open, type *CTRL+B*.
3. In a Windows terminal, navigate to your code folder. Type `./make` and press “Enter” to execute. This command searches the current directory for a makefile and then builds your program.

4. When the build is successful, you will see text at the bottom of the makefile like this:

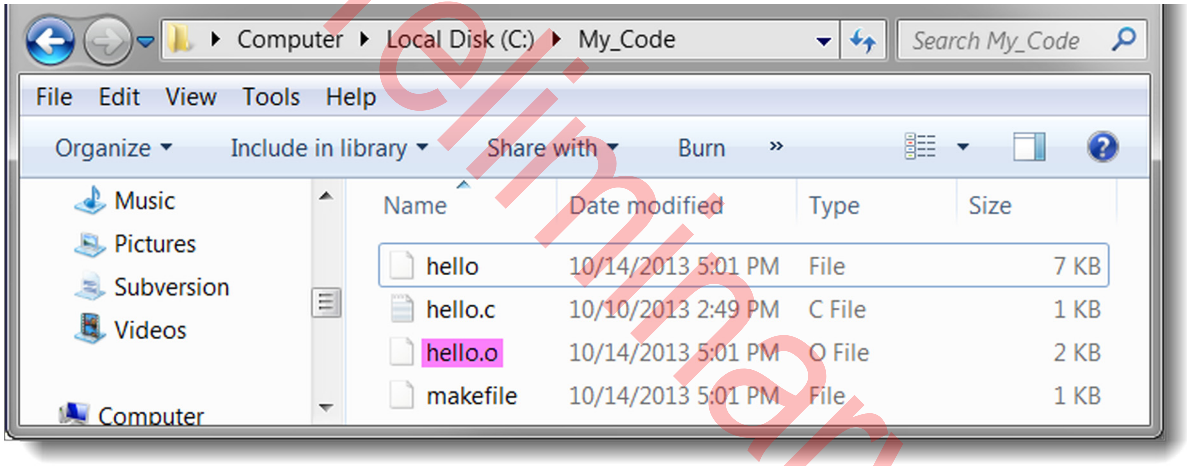

C:\My_Code\makefile - Sublime Text 2 (UNREGISTE...
File Edit Selection Find View Goto Tools Project Preferences Help
hello.c
makefile

```
CC=arm-none-linux-gnueabi-gcc
LD=arm-none-l     -gnueabi-gcc
 LAGS=-W
```

```
arm-none-linux-gnueabi-gcc hello.c -c -Wall
arm-none-linux-gnueabi-gcc hello.o -o hello
[Finished in 1.5s]
```

10 lines, 178 characters selected Tab Size: 4 Makefile

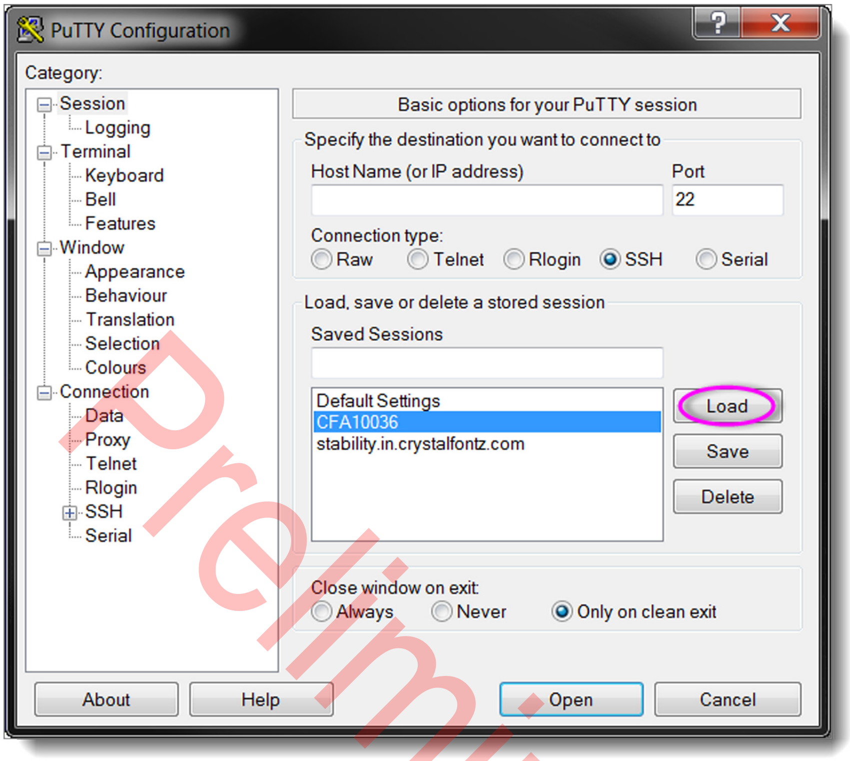
If you look in your *My_Code* folder, you will see that a *hello.o* file was created.


Computer ▸ Local Disk (C:) ▸ My_Code
Search My_Code
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ Burn »
Music
Pictures
Subversion
Videos
Computer

Name	Date modified	Type	Size
hello	10/14/2013 5:01 PM	File	7 KB
hello.c	10/10/2013 2:49 PM	C File	1 KB
hello.o	10/14/2013 5:01 PM	O File	2 KB
makefile	10/14/2013 5:01 PM	File	1 KB

USE PUTTY TO COMMUNICATE BETWEEN THE CFA10036 AND YOUR PC

1. Double click on the *PuTTY* shortcut you created earlier. The *PuTTY Configuration* window opens.
2. In the *Saved Session* list, look for the *CFA10036* session that you created earlier. Select it and click on the "Load" button.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

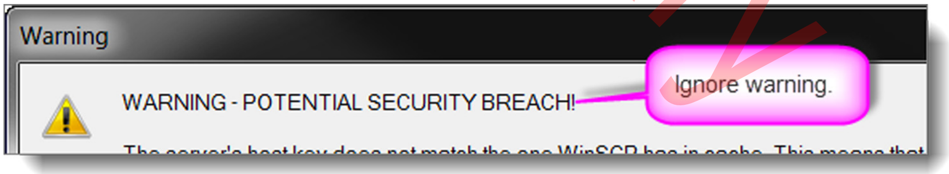
Load, save or delete a stored session

Saved Sessions

Default Settings
CFA10036
stability.in.crystalfontz.com

Close window on exit: ○ Always ○ Never ● Only on clean exit

1. Click on the *PuTTY* shortcut. The *PuTTY Configuration* window will open.
2. Under *Saved Sessions*, find *CFA10036* and select it.
3. Click on the "Load" button.
   A window opens with the heading "WARNING - POTENTIAL SECURITY BREACH! It is OK to ignore this warning by clicking on the "Yes" button.



WARNING - POTENTIAL SECURITY BREACH!

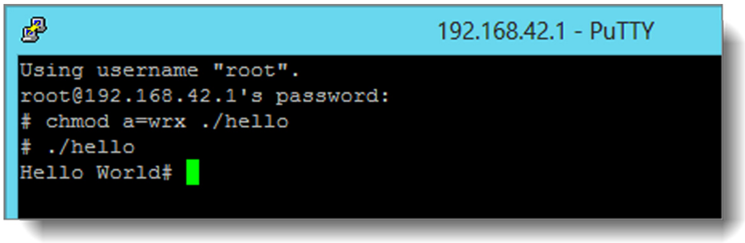
The server's host key does not match the one WinSCP has in cache. This means that

4. The program will try to connect to this IP address. You will be prompted to enter the user name, and password. Use:
   login: `root↵`
   password: `root↵`
5. To set the user permissions for ./hello, copy and paste this *chmod* command:

```
chmod a=wrx ./hello↵
```

To run the program, type:`./hello↵`

The result should be similar to this:



```
Using username "root".
root@192.168.42.1's password:
# chmod a=wrx ./hello
# ./hello
Hello World#
```

DISPLAY “HELLO WORLD” PROGRAM ON THE OPTIONAL OLED

THIS SECTION IS INCOMPLETE AND HAS NOT BEEN FULLY VALIDATED.

This section has instructions on how to run the “Hello World” program that you built on your PC to run on the optional OLED CFAL12832D-B. If you order the OLED at the same time as the CFA10036 when making your “Customize and Add to Cart” choices, we will install the OLED

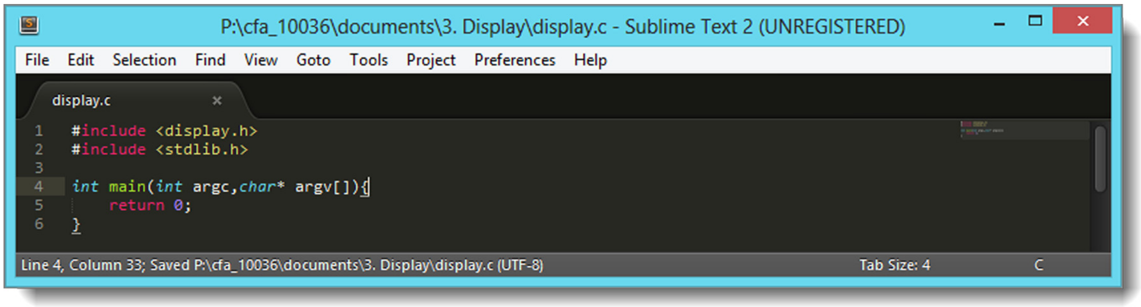
> Note
>
> You will need the *cfa-10036-documentation* files that are stored in the GitHub repository github.com/crystalfontz/cfa_10036_kernel). If you skipped the instructions in WINDOWS: DOWNLOAD SAMPLE CODE FROM GITHUB REPOSITORY (Pg. 43), go back and follow the download instructions first.

Create The Program

1. Make a new Sublime Text file named *display.c* and save it in your My_Code folder.
2. Copy and paste this text into the file. Save and close it.

```
#include <display.h>
#include <stdlib.h>

int main(int argc,char* argv[]){
   return 0
```

The first thing to do is to include *display.h* and write the main function

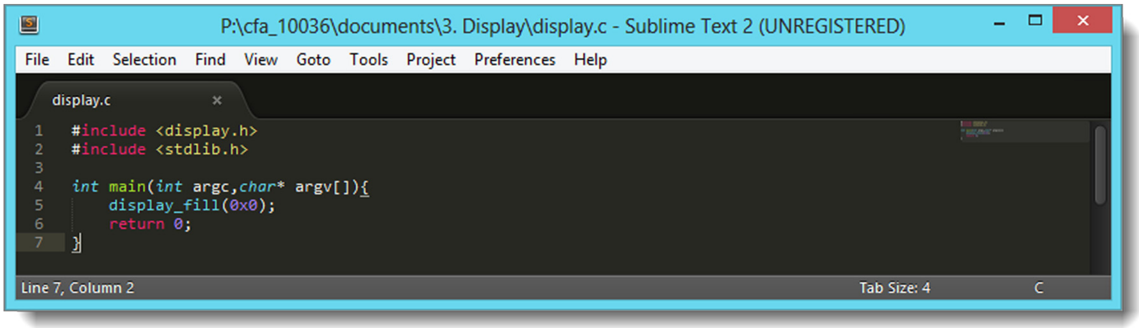

P:\cfa_10036\documents\3. Display\display.c - Sublime Text 2 (UNREGISTERED)
File
Edit
Selection
Find
View
Goto
Tools
Project
Preferences
Help
display.c

```
#include <display.h>
#include <stdlib.h>

int main(int argc,char* argv[]){
    return 0;
}
```

Line 4, Column 33; Saved P:\cfa_10036\documents\3. Display\display.c (UTF-8)   Tab Size: 4   C

Use The Library

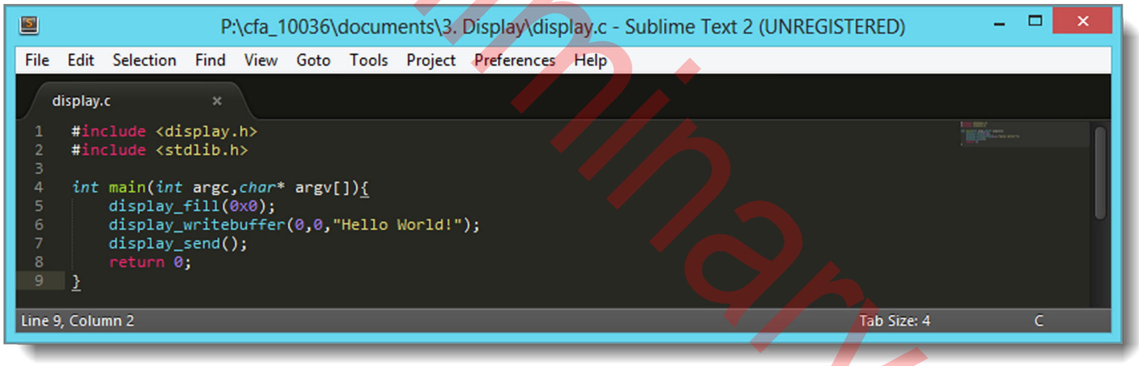
The *display.h* library is fairly simple with only three functions at the moment. The first one is *void display_fill(unsigned char byte*). *Display_fill* is used to fill the internal buffer with a specific byte value. It is mostly used to clear the buffer by calling it like so *display_fill(0x0);*.



```c
#include <display.h>
#include <stdlib.h>

int main(int argc,char* argv[]){
    display_fill(0x0);
    return 0;
}
```

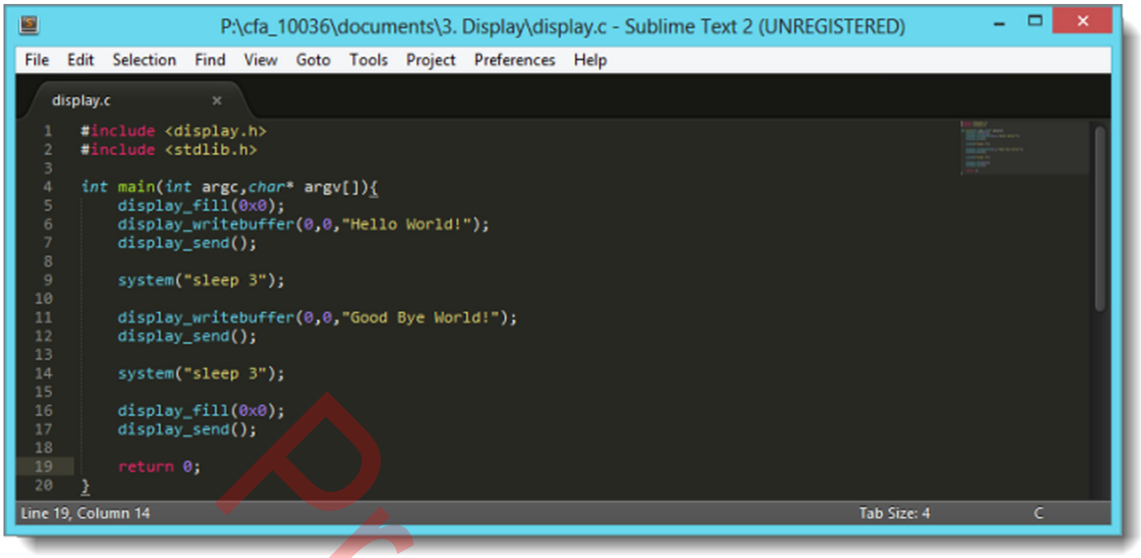
We are going to write to the buffer. We do this by calling `void display_writebuffer(int row, int col, char *string)`. This function allows us to write to the buffer. The first two parameters are the x and y coordinates that the message is printed at. If both row, and col are *0* (zero), then the string will be written at the top left of the screen. If row is *1*, the text will be shifted down by one row. String is the text that will be written to the buffer. The buffer doesn't take care of wordwrap. All it provides is a small layer of abstraction.

The first thing that we are going to write to the screen is Hello World. We are going to do that by calling `display_writebuffer(0,0,”Hello World!”);`. Doing this alone won't make any text show up on the screen. In order to observe any changes, we need to send the buffer out to the display by `calling void display_send(void):`.



```c
#include <display.h>
#include <stdlib.h>

int main(int argc,char* argv[]){
    display_fill(0x0);
    display_writebuffer(0,0,"Hello World!");
    display_send();
    return 0;
}
```

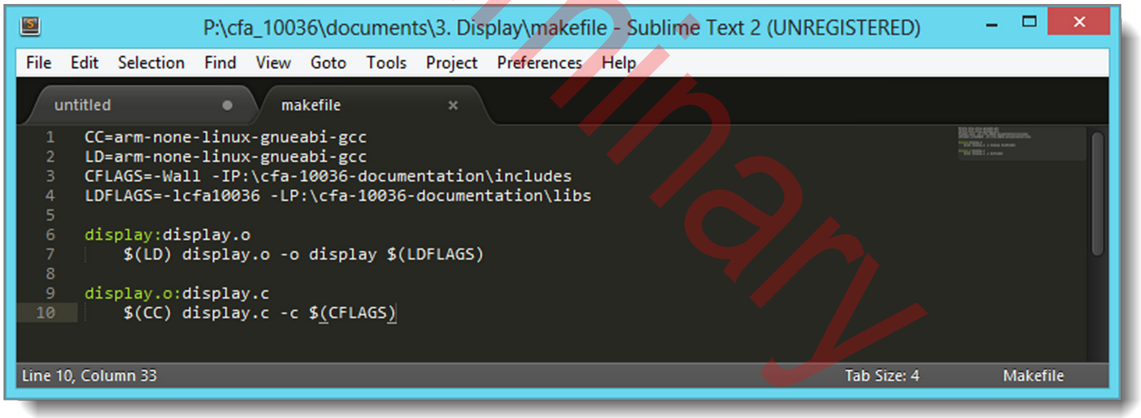
To finish our program, we are going to add something else: a little pause and then `Good Bye World`! We will accomplish this by calling `system("sleep 3")`, writing to the buffer, and then sending it out again. After we write Good Bye we will wait a few seconds and then clear the screen.


P:\cfa_10036\documents\3. Display\display.c - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
display.c
Line 19, Column 14
Tab Size: 4
C

```c
#include <display.h>
#include <stdlib.h>

int main(int argc,char* argv[]){
    display_fill(0x0);
    display_writebuffer(0,0,"Hello World!");
    display_send();

    system("sleep 3");

    display_writebuffer(0,0,"Good Bye World!");
    display_send();

    system("sleep 3");

    display_fill(0x0);
    display_send();

    return 0;
}
```

Compile The Program

All that is left is to compile, and run. To compile we will create a *makefile* similar to what we used before in the Hello World guide. The only addition to it is that I will change the some `cflag` and `ldflag`.


P:\cfa_10036\documents\3. Display\makefile - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
untitled
makefile
Line 10, Column 33
Tab Size: 4
Makefile

```makefile
CC=arm-none-linux-gnueabi-gcc
LD=arm-none-linux-gnueabi-gcc
CFLAGS=-Wall -IP:\cfa-10036-documentation\includes
LDFLAGS=-lcfa10036 -LP:\cfa-10036-documentation\libs

display:display.o
    $(LD) display.o -o display $(LDFLAGS)

display.o:display.c
    $(CC) display.c -c $(CFLAGS)
```

If you look at the cflags and ldflags variables you will notice that I added in a library search path and a header search path so that the compiler and linker will be able to find display.h and link against the header files. You will need to replace the P:\*\ with t6he directory that you installed the library into. Enter the directory and run make.

From here all that needs to be done is to install it onto the CFA10036 by using WinSCP, and then run the program by using Putty.

LINUX: DEVELOPMENT TOOLS INSTALLATION

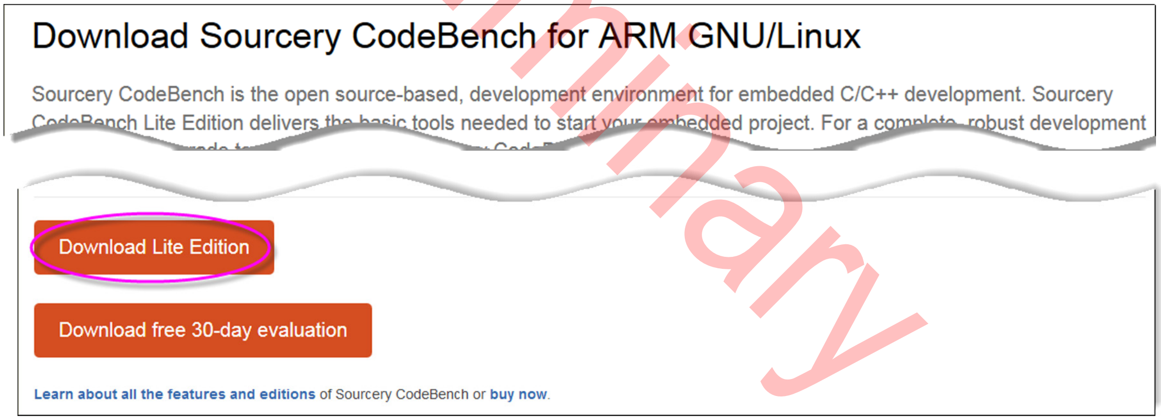
> Note
>
> For the Linux instructions below, we used *Ubuntu 12.04 LTS*. This long-term support release has guaranteed security and support updates until April 2017.
>
> If you use a different Linux operating system (for example, Red Hat), the steps below may be different.

CROSS COMPLIER

You will need a cross compiler program. We used the free *Sourcery CodeBench Lite arm-2013.05-24-arm-none-linux-gnueabi.exe* from Mentor Graphics.

> Note
>
> Mentor Graphics has several versions of Sourcery CodeBench Lite with slightly different names. To ensure you have the program that works with these instructions, You need to download and install the file *arm-DATE-arm-none-linux-gnueabi.exe*. Carefully follow the steps below.

1. Go to https://sourcery.mentor.com/GNUToolchain/release2450.
2. The web page *Download Sourcery CodeBench for ARM GNU/Linux* will display. Click on the orange button “Download Lite Edition” near the bottom of the page.



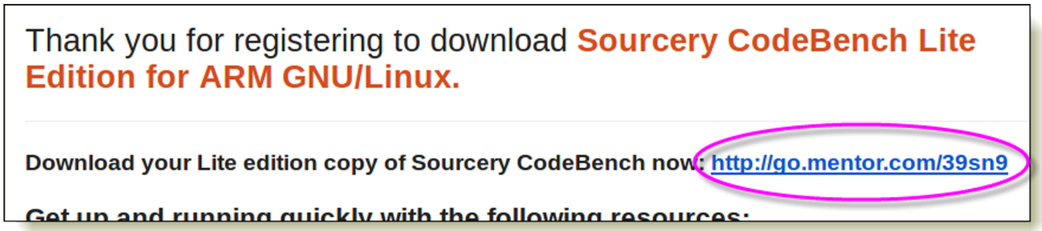
Download Sourcery CodeBench for ARM GNU/Linux

Sourcery CodeBench is the open source-based, development environment for embedded C/C++ development. Sourcery CodeBench Lite Edition delivers the basic tools needed to start your embedded project. For a complete, robust development

Learn about all the features and editions of Sourcery CodeBench or buy now.

3. If you are not already signed in to Mentor Graphics, a web page will open where you will be prompted to create and activate a free account.
4. Create and sign in to your account. A web page will display with the heading *Check your E-mail for Download Link*.

5. Find and open the email from *sourcery_support*. The email acknowledges your registration and includes a unique link to download the program.

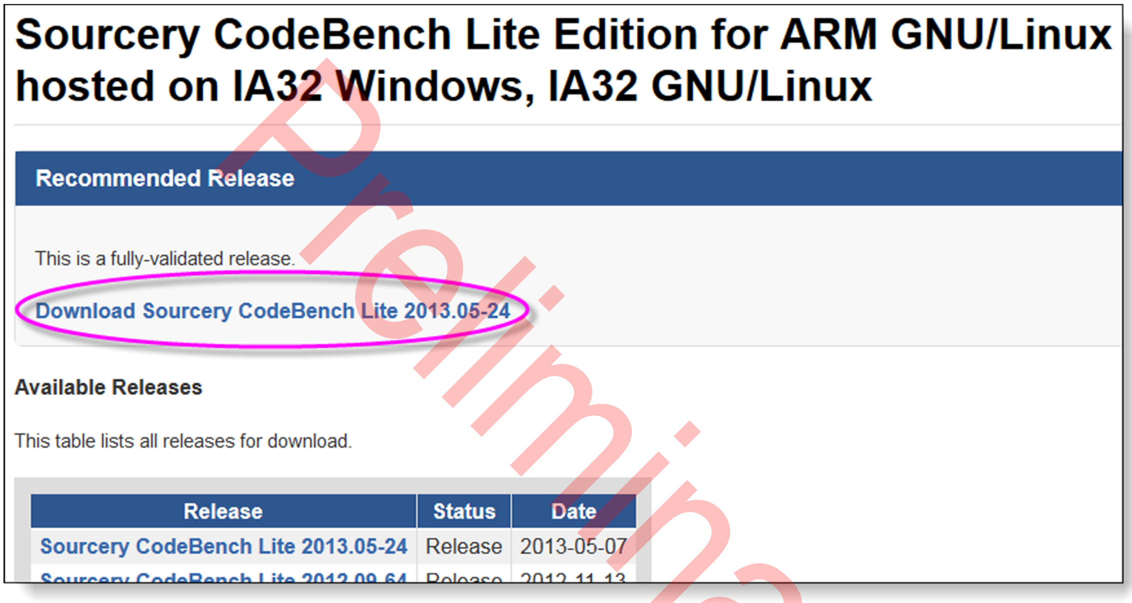


Thank you for registering to download **Sourcery CodeBench Lite Edition for ARM GNU/Linux.**

Download your Lite edition copy of Sourcery CodeBench now: http://go.mentor.com/39sn9

Get up and running quickly with the following resources:

6. A web page with the heading *Sourcery CodeBench Lite Edition for ARM GNU/Linux hosted on IA32 Windows, IA32 GNU/Linux* will display. You will see a list of downloads. Under the *Recommended Release* subheading, click on the “Download Sourcery CodeBench Lite 2013-.05-24”.



Sourcery CodeBench Lite Edition for ARM GNU/Linux hosted on IA32 Windows, IA32 GNU/Linux

Recommended Release

This is a fully-validated release.

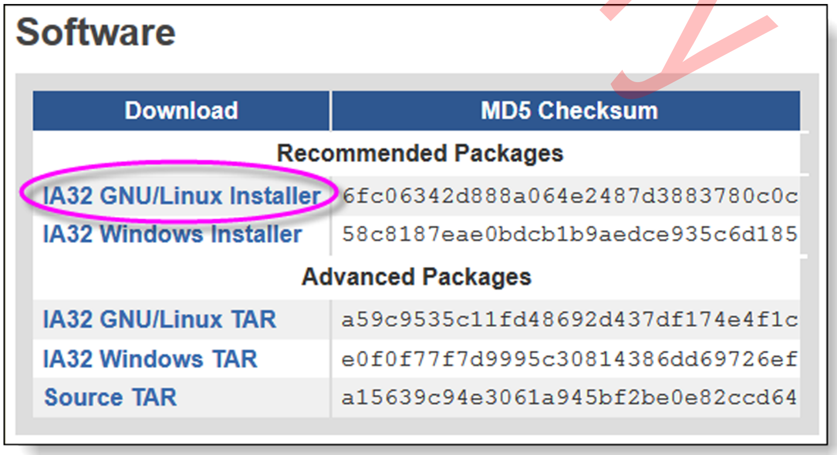
Download Sourcery CodeBench Lite 2013.05-24

Available Releases

This table lists all releases for download.

Release	Status	Date
Sourcery CodeBench Lite 2013.05-24	Release	2013-05-07
Sourcery CodeBench Lite 2012.09-64	Release	2012-11-13

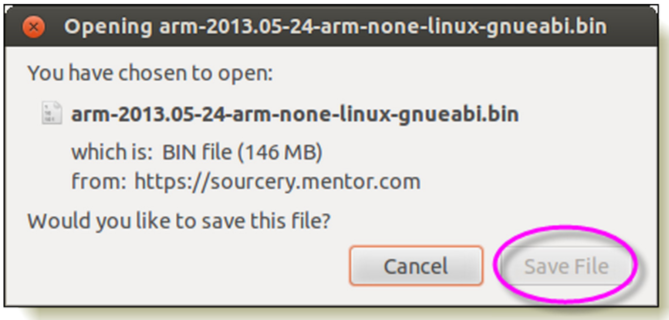
7. A page with the heading *Sourcery CodeBench Lite 2013.05-24* will display. Under the *Software* heading, you will see a list of downloads. Select the “IA32 GNU/Linux Installer”.



Software

Download	MD5 Checksum
Recommended Packages	
IA32 GNU/Linux Installer	6fc06342d888a064e2487d3883780c0c
IA32 Windows Installer	58c8187eae0bdcb1b9aedce935c6d185
Advanced Packages	
IA32 GNU/Linux TAR	a59c9535c11fd48692d437df174e4f1c
IA32 Windows TAR	e0f0f77f7d9995c30814386dd69726ef
Source TAR	a15639c94e3061a945bf2be0e82ccd64

8. You are prompted to save a .bin file.



Opening arm-2013.05-24-arm-none-linux-gnueabi.bin

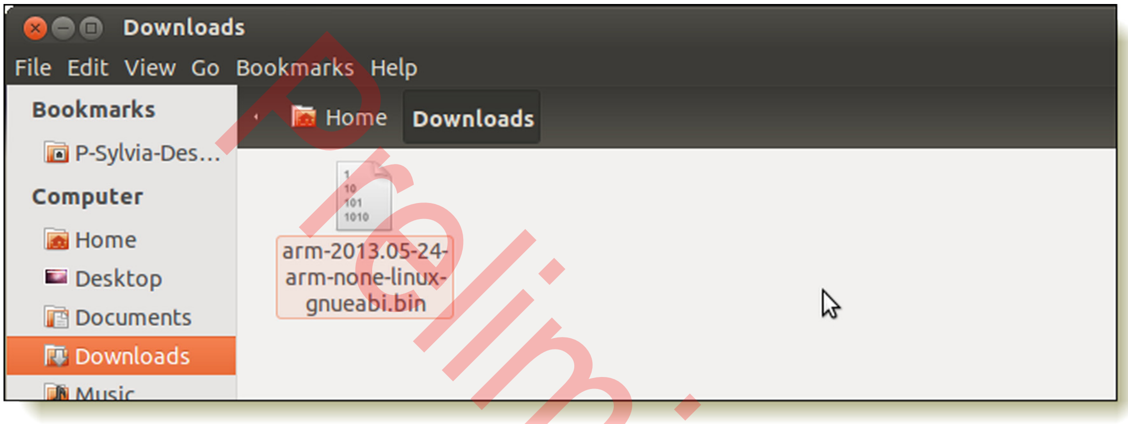
You have chosen to open:

arm-2013.05-24-arm-none-linux-gnueabi.bin

which is: BIN file (146 MB)
from: https://sourcery.mentor.com

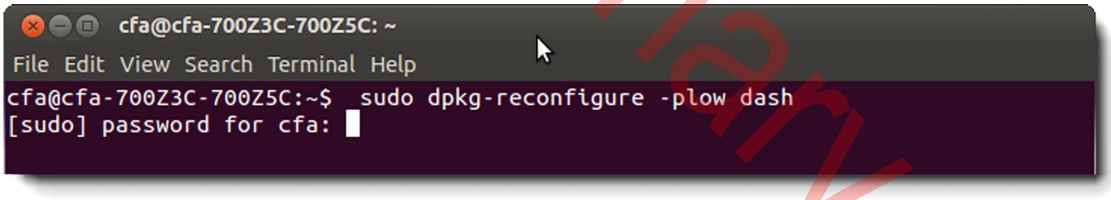
Would you like to save this file?

Save the file to your computer. Note the download location of the file. You will need to navigate to this location in a step below.



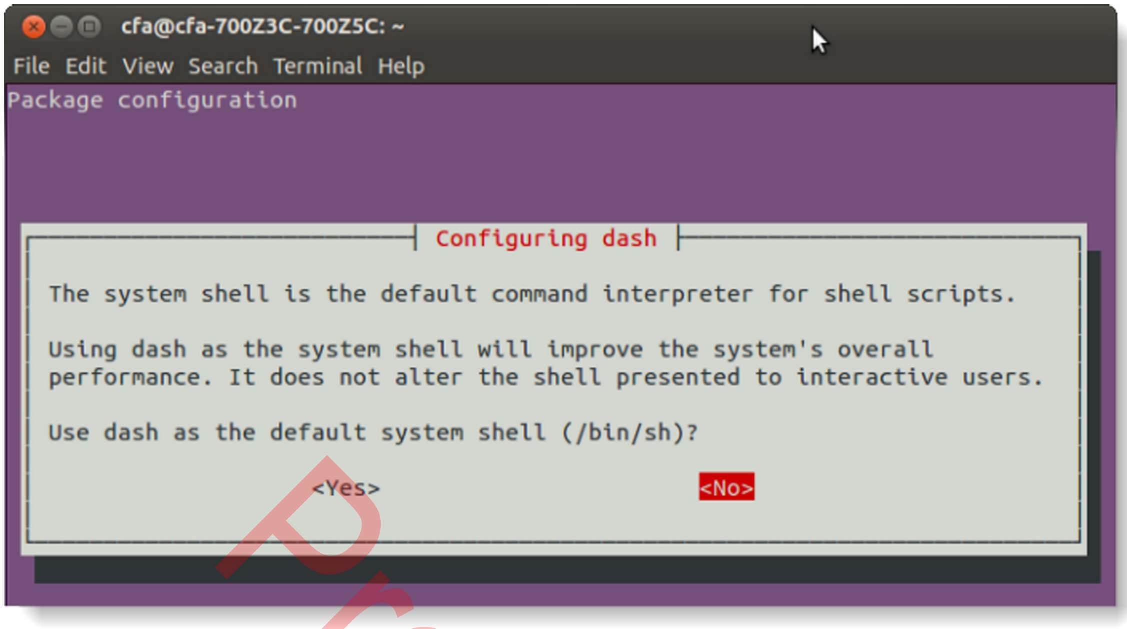
9. In a terminal window, execute the command `sudo dpkg-reconfigure -plow dash`

10. You will be prompted for your computer's user password.



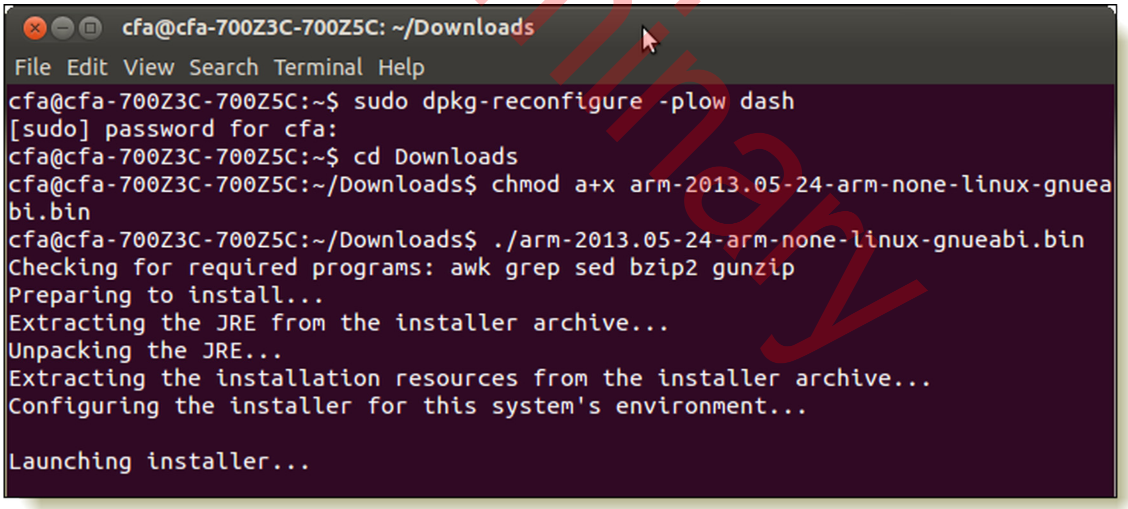
```
cfa@cfa-700Z3C-700Z5C:~$  sudo dpkg-reconfigure -plow dash
[sudo] password for cfa: 
```

11. You will be asked if “dash” should be the default shell. Choose the default reply “No”.



```
Package configuration

Configuring dash

The system shell is the default command interpreter for shell scripts.

Using dash as the system shell will improve the system's overall
performance. It does not alter the shell presented to interactive users.

Use dash as the default system shell (/bin/sh)?

        <Yes>                       <No>
```

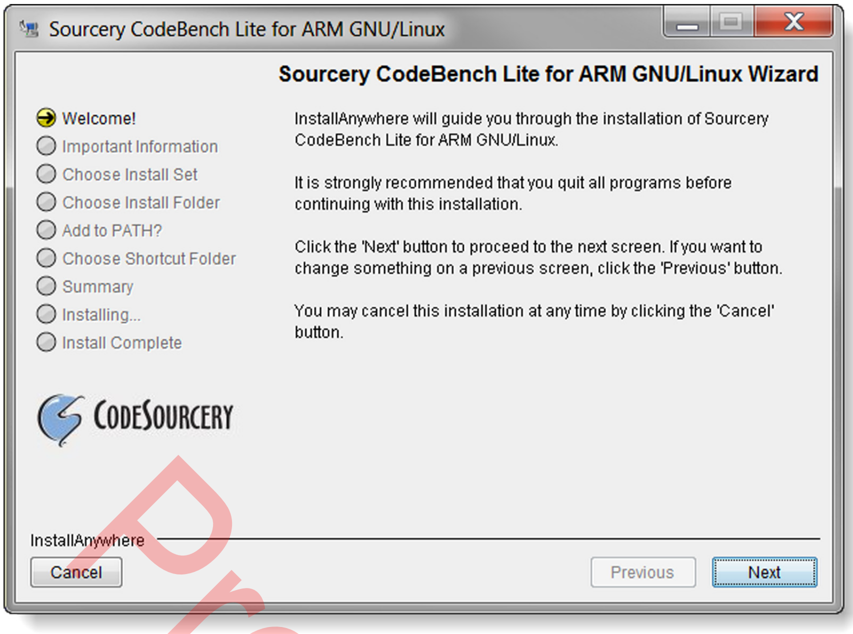
12. In the terminal window, Execute *cd* to navigate to the folder where you downloaded the .bin file. Example: `cd Downloads`
13. Execute `chmod a+x arm-2013.05-24-arm-none-linux-gnueabi.bin`
14. Execute: `./arm-2013.05-24-arm-none-linux-gnueabi.bin`
    The terminal window will look like this:



cfa@cfa-700Z3C-700Z5C: ~/Downloads
File Edit View Search Terminal Help

```
cfa@cfa-700Z3C-700Z5C:~$ sudo dpkg-reconfigure -plow dash
[sudo] password for cfa:
cfa@cfa-700Z3C-700Z5C:~$ cd Downloads
cfa@cfa-700Z3C-700Z5C:~/Downloads$ chmod a+x arm-2013.05-24-arm-none-linux-gnuea
bi.bin
cfa@cfa-700Z3C-700Z5C:~/Downloads$ ./arm-2013.05-24-arm-none-linux-gnueabi.bin
Checking for required programs: awk grep sed bzip2 gunzip
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...
```

15. The installation wizard *Sourcery CodeBenchLite for ARM GNU/Linux Wizard* will open.



Sourcery CodeBench Lite for ARM GNU/Linux

Sourcery CodeBench Lite for ARM GNU/Linux Wizard

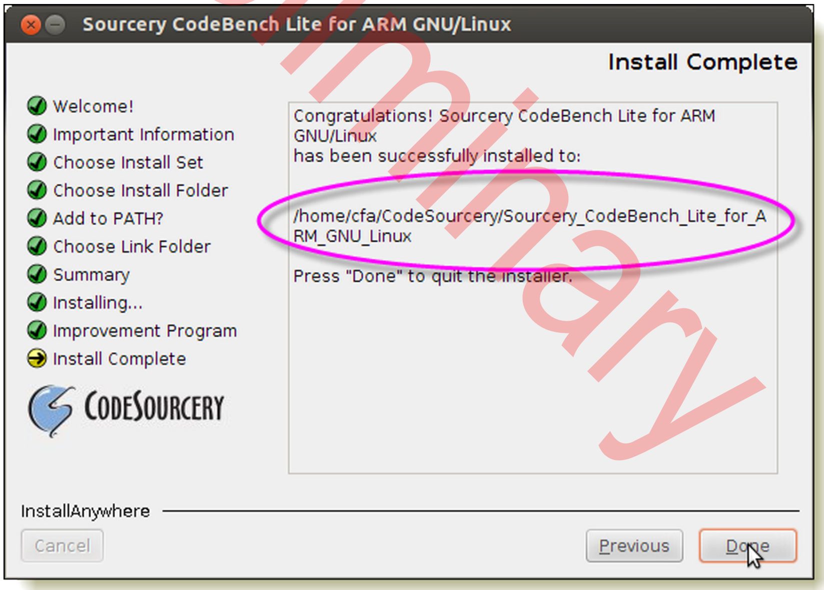
InstallAnywhere will guide you through the installation of Sourcery CodeBench Lite for ARM GNU/Linux.

It is strongly recommended that you quit all programs before continuing with this installation.

Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

16. Follow the steps in the wizard's instructions for a *Typical* installation. The default options are fine. Note the location of the installation.



Sourcery CodeBench Lite for ARM GNU/Linux

Install Complete

Congratulations! Sourcery CodeBench Lite for ARM GNU/Linux
has been successfully installed to:

/home/cfa/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux

Press "Done" to quit the installer.

SAMPLE CODE

External libraries are not required to program the CFA10036. There are free sample code files with documentation available on GitHub here: github.com/crystalfontz/cfa_10036_kernel). GitHub is an open source revision control system for software development projects.

Install Git

The program *Git* is required to download the CFA10036 sample code and documentation from the GitHub repository. The execution to download the Git program will differ, depending on if you want the 32-bit or 64-bit version.

1. *For Git 32-bit*

   Execute
   ```
   sudo apt-get install git build-essential zip gawk libtool gettext automake autoconf nasm unzip flex bison libsdl-image1.2 liblzo2-2 lzma python openjdk-7-jre-headless texinfo libxml-parser-perl wget pkg-config swig cpio zlib1g-dev subversion binutils
   ```

   For Git 64-bit

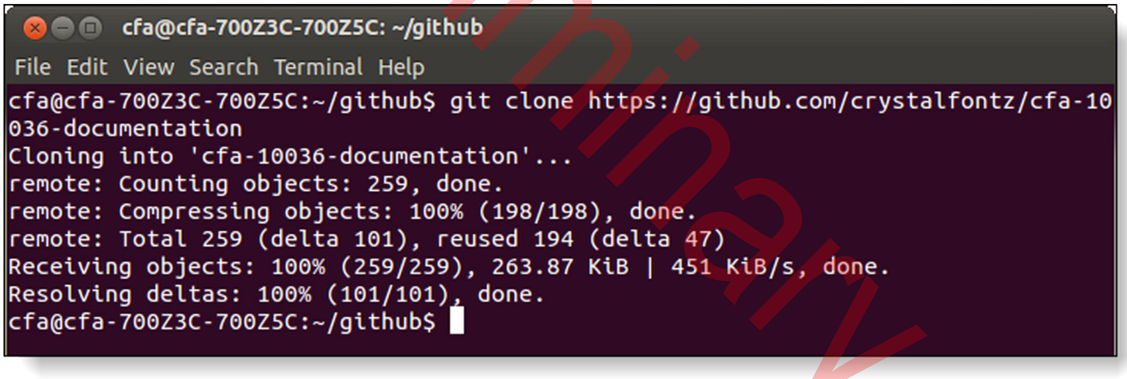
   Execute
   ```
   sudo apt-get install git build-essential zip gawk libtool gettext automake autoconf nasm unzip flex bison libsdl-image1.2 liblzo2-2 lzma python openjdk-7-jre-headless texinfo libxml-parser-perl wget pkg-config swig cpio zlib1g-dev subversion ia32-libs-gtk ia32-libs binutils
   ```

2. Enter your computer's user password at the prompt. Git will automatically download and install.

How To Download The CFA10036 Repository From GitHub

1. Create a working directory to clone the repository into. We recommend that you create the working directory in your Home folder.
2. Navigate to your working directory and execute this in a terminal window:
   ```
   git clone https://github.com/crystalfontz/cfa-10036-documentation
   ```

On some terminals, you may need to use SHIFT+INSERT keys instead of copy and paste.


cfa@cfa-700Z3C-700Z5C: ~/github
File Edit View Search Terminal Help

```
cfa@cfa-700Z3C-700Z5C:~/github$ git clone https://github.com/crystalfontz/cfa-10
036-documentation
Cloning into 'cfa-10036-documentation'...
remote: Counting objects: 259, done.
remote: Compressing objects: 100% (198/198), done.
remote: Total 259 (delta 101), reused 194 (delta 47)
Receiving objects: 100% (259/259), 263.87 KiB | 451 KiB/s, done.
Resolving deltas: 100% (101/101), done.
cfa@cfa-700Z3C-700Z5C:~/github$
```

Git will automatically download the most recent files and install them into a new folder named *cfa-10036-documentation* in the working (current) directory.

IF YOU WANT TO SET THE PATH MANUALLY

The installer gives you the option to set the path. If you would like to set path manually later, execute

```
echo 'export PATH=$PATH:/INSTALLATIONDIRECTORY' >> ~/.bashrc
```

Note: You will need to log out and log back in for the path set to take effect.

LINUX: WRITE AND COMPILE A PROGRAM “HELLO WORLD”

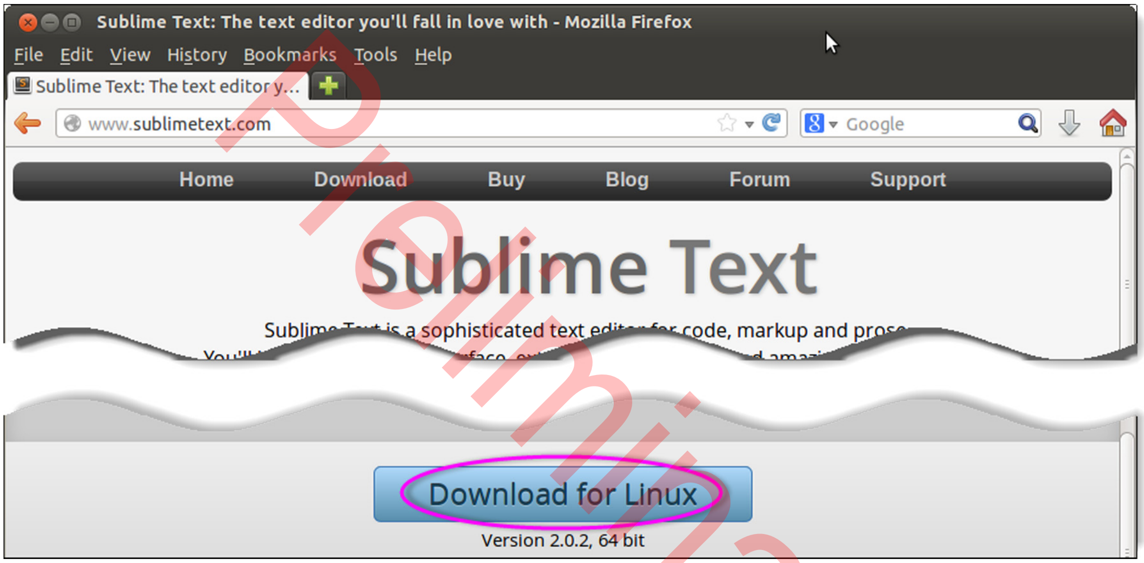
THIS SECTION IS INCOMPLETE AND HAS NOT BEEN FULLY VALIDATED.

This section describes how to write, compile, and upload a simple program on the CFA10036.

LINUX: OBTAIN A TEXT EDITOR

To write our Hello World program, we used the text editor *Sublime Text 2* (As of 2013-10-10, a beta version of Sublime Text 3 is also available).

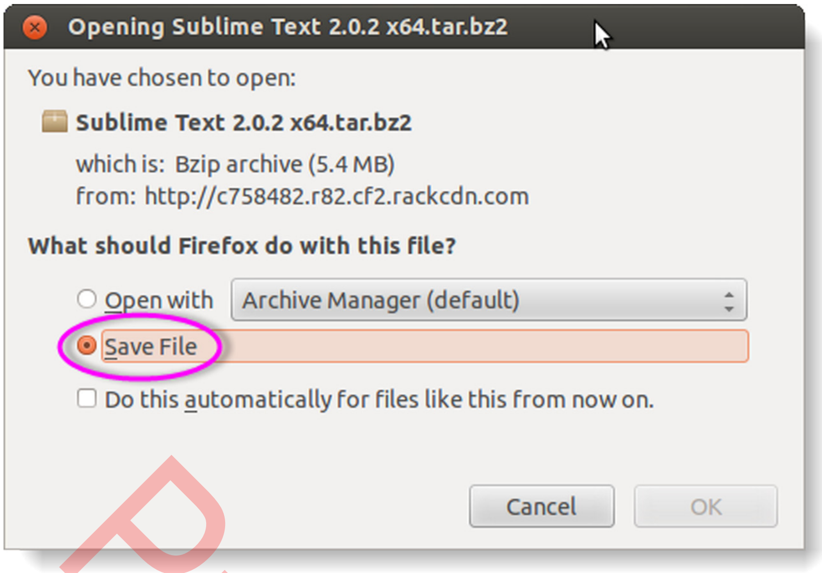
1. Go to www.sublimetext.com. The suggested download for your operating system will be shown as a blue button.


Sublime Text: The text editor you'll fall in love with - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Sublime Text: The text editor y...
www.sublimetext.com
Google
Home
Download
Buy
Blog
Forum
Support
Sublime Text
Sublime Text is a sophisticated text editor for code, markup and prose
Download for Linux
Version 2.0.2, 64 bit

When your mouse arrow hovers over the button, a popup with a link to other platforms displays.

This free program has no enforced time limit, but occasionally has a popup display that requests that you purchase the full version (~$70 in October 2013).

2. Click on the “Download for Linux” button. You will be prompted to save *Sublime Text VERSION.tar.bz2.*



Opening Sublime Text 2.0.2 x64.tar.bz2

You have chosen to open:

Sublime Text 2.0.2 x64.tar.bz2

which is: Bzip archive (5.4 MB)
from: http://c758482.r82.cf2.rackcdn.com

What should Firefox do with this file?

○ Open with Archive Manager (default)
● Save File
□ Do this automatically for files like this from now on.

Save the file to your computer. Note the location where you saved the file.

3. Go to the *Sublime Text VERSION.tar.bz2* and double click on it to open the folder. Select “Extract” from the top of the menu.
4. Open the folder and find *sublime_text,* an executable file. Make a *Sublime Text* shortcut in a convenient location: right-click and select “Make Link” from the dropdown menu. Or select “Bookmark>Add” a Bookmark to add a Bookmark to the Ubuntu browser.

5. Click on the *Sublime Text* shortcut or bookmark. A window will open that looks similar to a terminal window.

LINUX: WRITE THE HELLO WORLD CODE

The following instructions will show you how to write is a simple “Hello World” program in C.

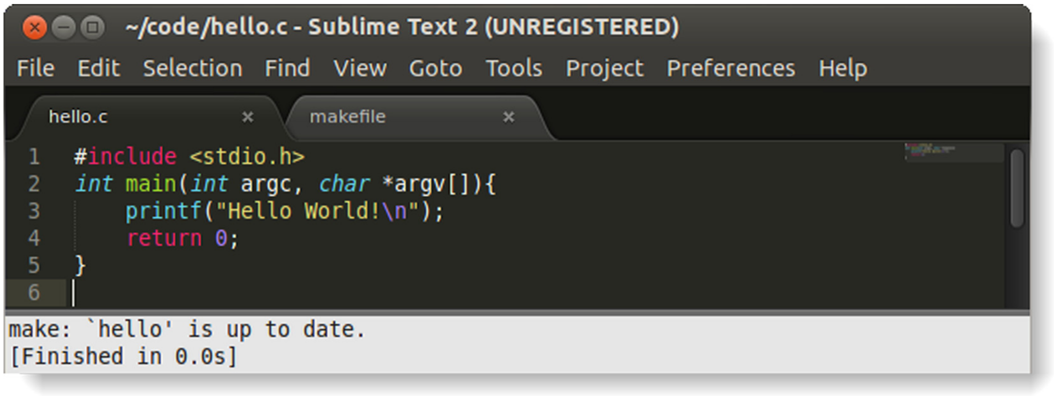
1. Create a folder close to your working directory to hold the files for your program. Example:

Add image here.

2. In Sublime Text, use File>New to create a new empty window.
3. Copy and paste the text below into the window.:

```
#include <stdio.h>
int main(int argc, char *argv[]){
printf("Hello World!\n");
```

```
return 0;
```



```
#include <stdio.h>
int main(int argc, char *argv[]){
    printf("Hello World!\n");
    return 0;
}

```

```
make: `hello' is up to date.
[Finished in 0.0s]
```

If colors are not automatically applied to the text, click on the *Plain Text* button at the bottom of the window., A dropdown list will display. Select “C*”* to add colors.

4. Save this Sublime Text file in your working directory as *hello.c*.The .c extension represents C code.

LINUX: COMPILE THE HELLO WORLD CODE

5. Now that you’ve written the hello world program, it needs to be compiled using *GNU make*. Make will compile the C code file into objects (.o extensions) and link them all together with the required libraries at the end. To understand this process, we suggest that you look at the make documentation and examples here: www.gnu.org/software/make/manual/make.html.
6. Using Sublime Text, create a new file and save it as *makefile* with no extension in your working directory, the same location as the hello.c file. Copy & paste this text into the new makefile:

```
CC=arm-none-linux-gnueabi-gcc
LD=arm-none-linux-gnueabi-gcc
CFLAGS=-Wall
LDFLAGS=

hello:hello.o
$(LD) hello.o -o hello $(LDFLAGS)

hello.o:hello.c
$(CC) hello.c -c $(CFLAGS)
```

To compile the program, type the following:

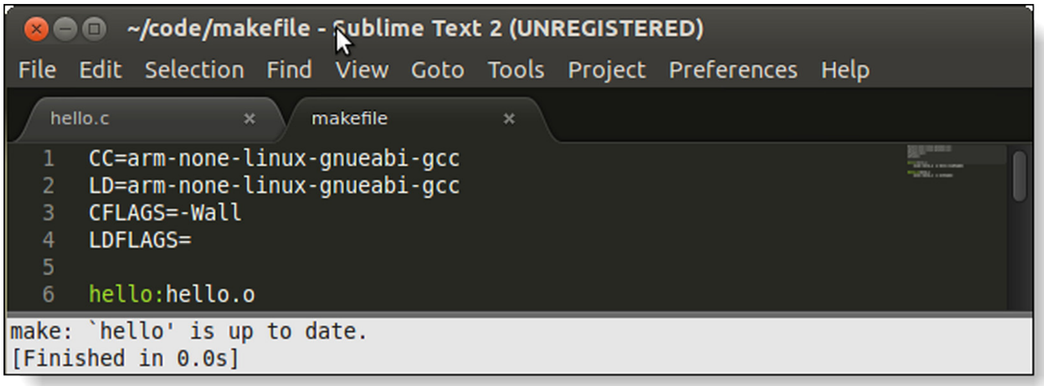
```
arm-none-linux-gnueabi-gcc hello.c -o hello
```

This should create a file called “`hello`” in your directory.
Using WinSCP (SCP under Linux), copy the `hello` file over to the CFA10036.

Typing `/hello` from a terminal connected to the CFA10036 should produce the following:

```
Welcome to the CFA10036 environment!
```

7. The Build System needs to be set to Make by clicking on Tools>Build System>Make. ?Each time you want to compile click Tools? Build or type Ctrl+B.



```
CC=arm-none-linux-gnueabi-gcc
LD=arm-none-linux-gnueabi-gcc
CFLAGS=-Wall
LDFLAGS=

hello:hello.o
```

```
make: `hello' is up to date.
[Finished in 0.0s]
```

APPENDIX A: HOW TO INSTALL UBUNTU ON WINDOWS 8 USING VMWARE PLAYER

THIS APPENDIX IS INCOMPLETE AND HAS NOT BEEN FULLY VALIDATED.

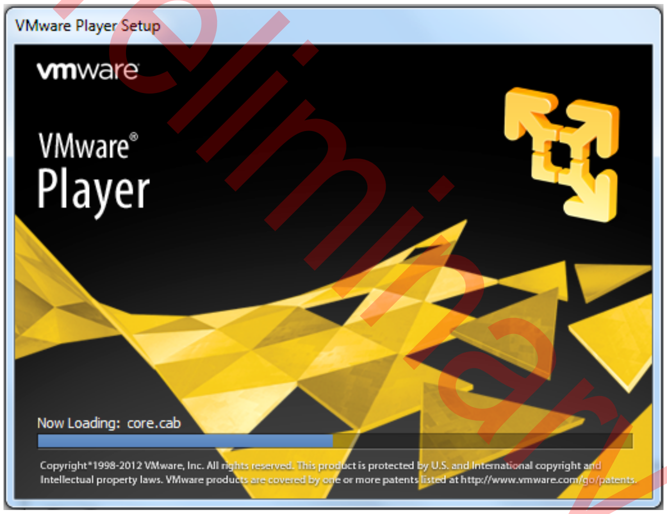
INTRODUCTION

To run a Linux distribution under the Windows 7 or Windows 8, it is necessary to first install virtual machine software to install and run the Linux. This document describes how to install the VMware Player software.

VMware Player was tested under Windows 7 and Windows 8. The player can be downloaded from www.vmware.com/products/player. The download is free for noncommercial use.

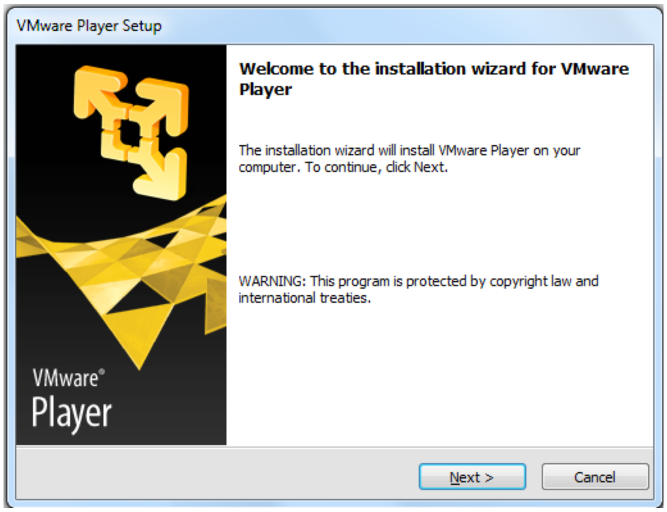
STEP 1: INSTALL VMPLAYER

Installation is straightforward and works well with the default settings. The following screenshots show the installation sequence for VMware Player.


VMware Player Setup

Now Loading: core.cab

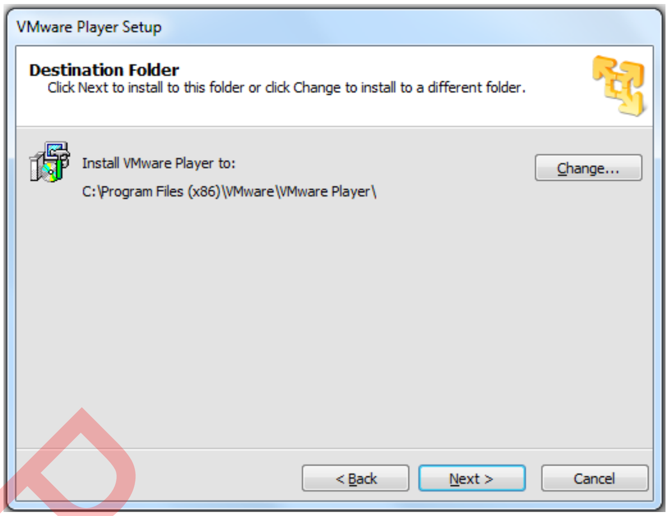
Copyright*1998-2012 VMware, Inc. All rights reserved. This product is protected by U.S. and International copyright and Intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents.


VMware Player Setup

Welcome to the installation wizard for VMware Player

The installation wizard will install VMware Player on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

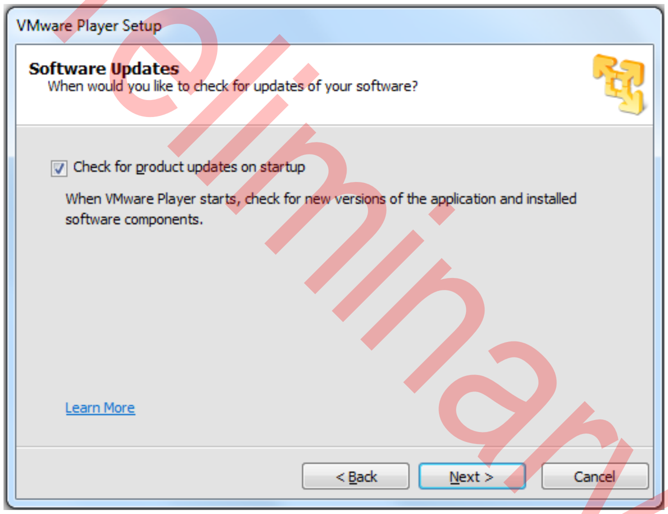

VMware Player Setup

Destination Folder

Click Next to install to this folder or click Change to install to a different folder.

Install VMware Player to:

C:\Program Files (x86)\VMware\VMware Player\

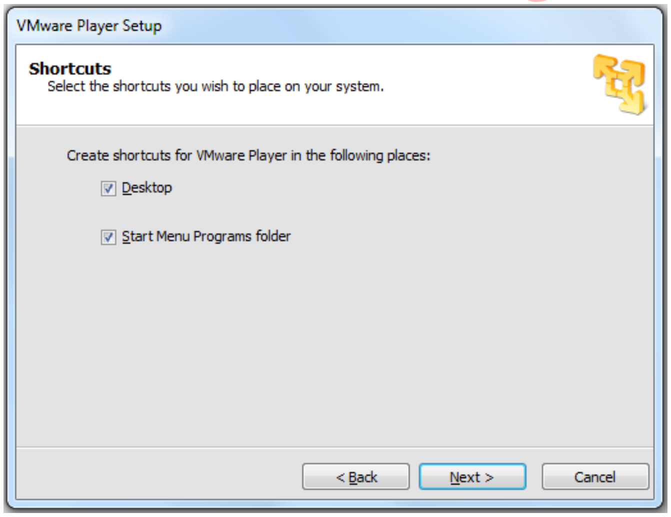

VMware Player Setup

Software Updates

When would you like to check for updates of your software?

☑ Check for product updates on startup

When VMware Player starts, check for new versions of the application and installed software components.


VMware Player Setup

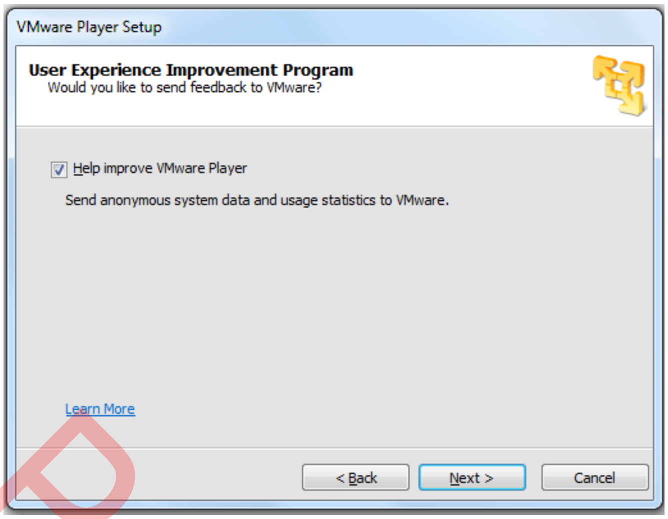
Shortcuts

Select the shortcuts you wish to place on your system.

Create shortcuts for VMware Player in the following places:

☑ Desktop

☑ Start Menu Programs folder



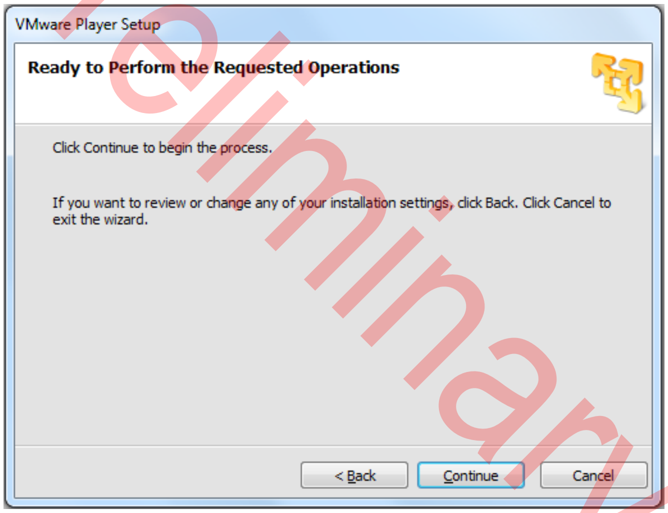
User Experience Improvement Program

Would you like to send feedback to VMware?

☑ Help improve VMware Player

Send anonymous system data and usage statistics to VMware.

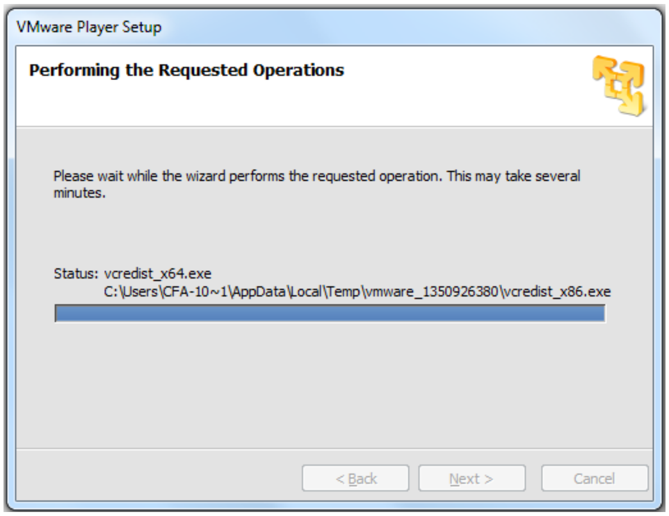
Learn More



Ready to Perform the Requested Operations

Click Continue to begin the process.

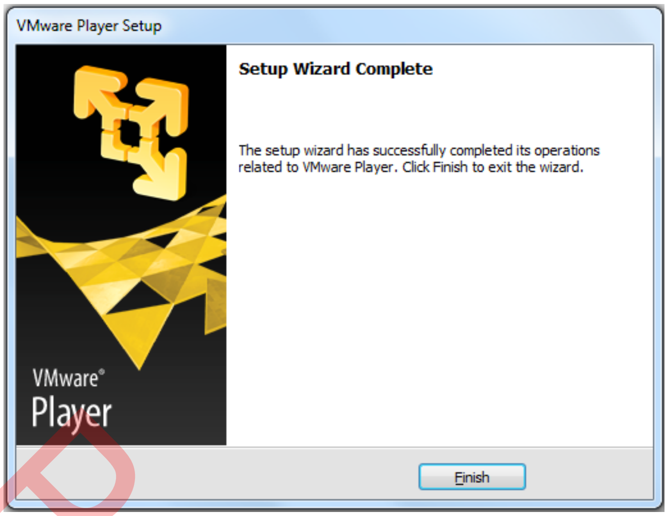
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.



Performing the Requested Operations

Please wait while the wizard performs the requested operation. This may take several minutes.

Status: vcredist_x64.exe
C:\Users\CFA-10~1\AppData\Local\Temp\vmware_1350926380\vcredist_x86.exe



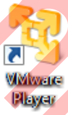
Setup Wizard Complete

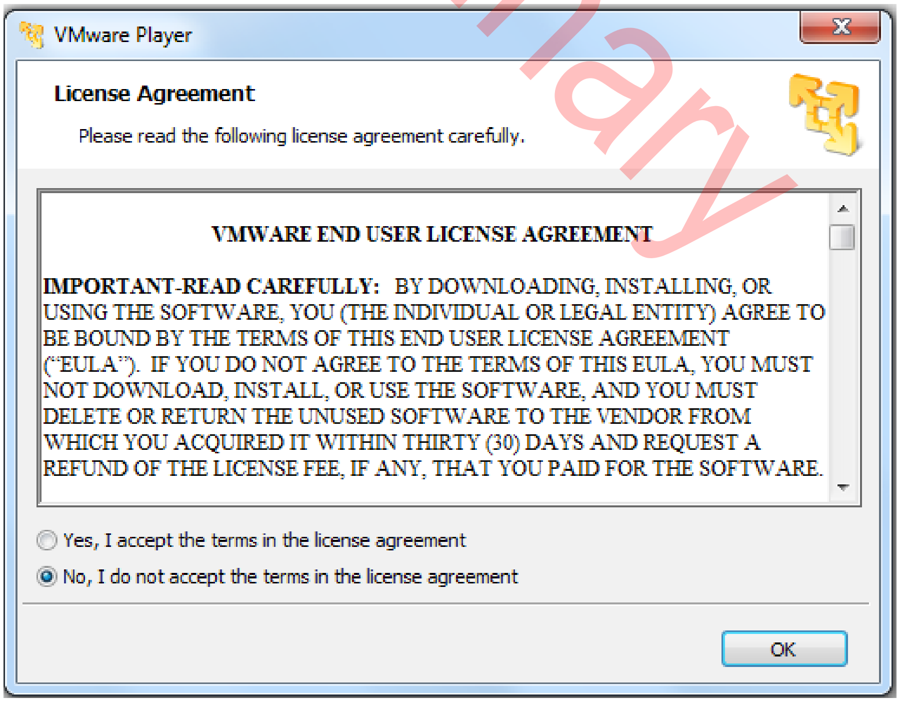
The setup wizard has successfully completed its operations related to VMware Player. Click Finish to exit the wizard.

VMware should now be fully installed and ready to use.

STEP 2: INSTALL UBUNTU

Insert the Ubuntu installation media into the PC's DVD player and click on the desktop icon to start the VMware Player.





License Agreement

Please read the following license agreement carefully.

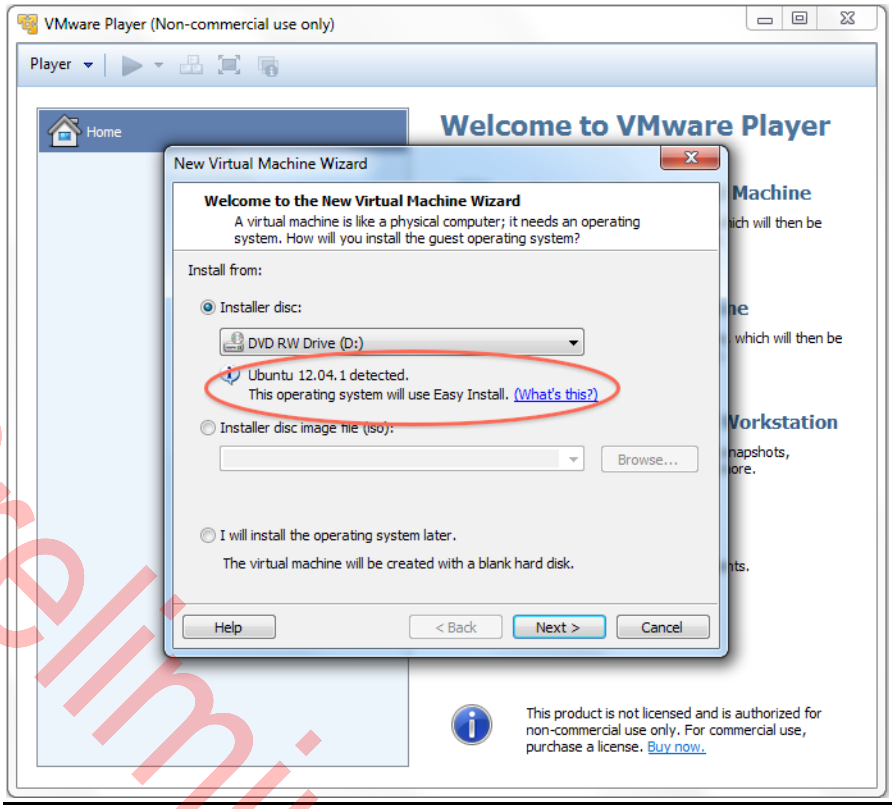
VMWARE END USER LICENSE AGREEMENT

IMPORTANT-READ CAREFULLY: BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE, YOU (THE INDIVIDUAL OR LEGAL ENTITY) AGREE TO BE BOUND BY THE TERMS OF THIS END USER LICENSE AGREEMENT ("EULA"). IF YOU DO NOT AGREE TO THE TERMS OF THIS EULA, YOU MUST NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND YOU MUST DELETE OR RETURN THE UNUSED SOFTWARE TO THE VENDOR FROM WHICH YOU ACQUIRED IT WITHIN THIRTY (30) DAYS AND REQUEST A REFUND OF THE LICENSE FEE, IF ANY, THAT YOU PAID FOR THE SOFTWARE.

○ Yes, I accept the terms in the license agreement

● No, I do not accept the terms in the license agreement

The virtual machine wizard should detect the Ubuntu as shown here.



Welcome to the New Virtual Machine Wizard

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

● Installer disc:

DVD RW Drive (D:)

Ubuntu 12.04.1 detected.
This operating system will use Easy Install. (What's this?)

○ Installer disc image file (iso):

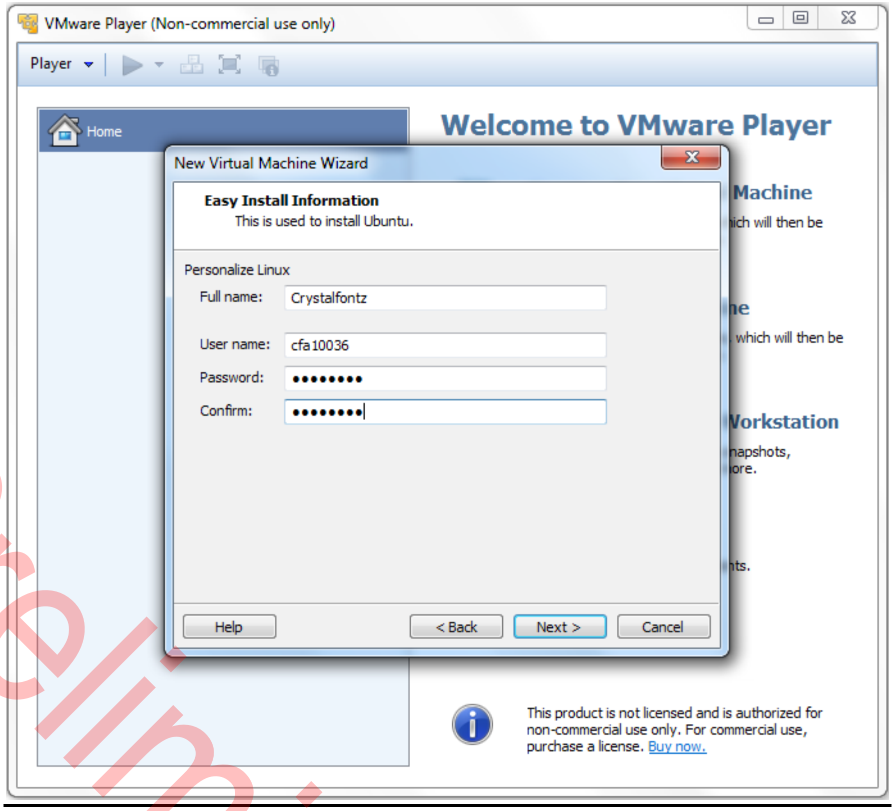
Browse...

○ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Enter the name you wish to give the virtual machine along with a user ID and password. This information will be passed to the Ubuntu installer.



Easy Install Information

This is used to install Ubuntu.

Personalize Linux

Full name: Crystalfontz

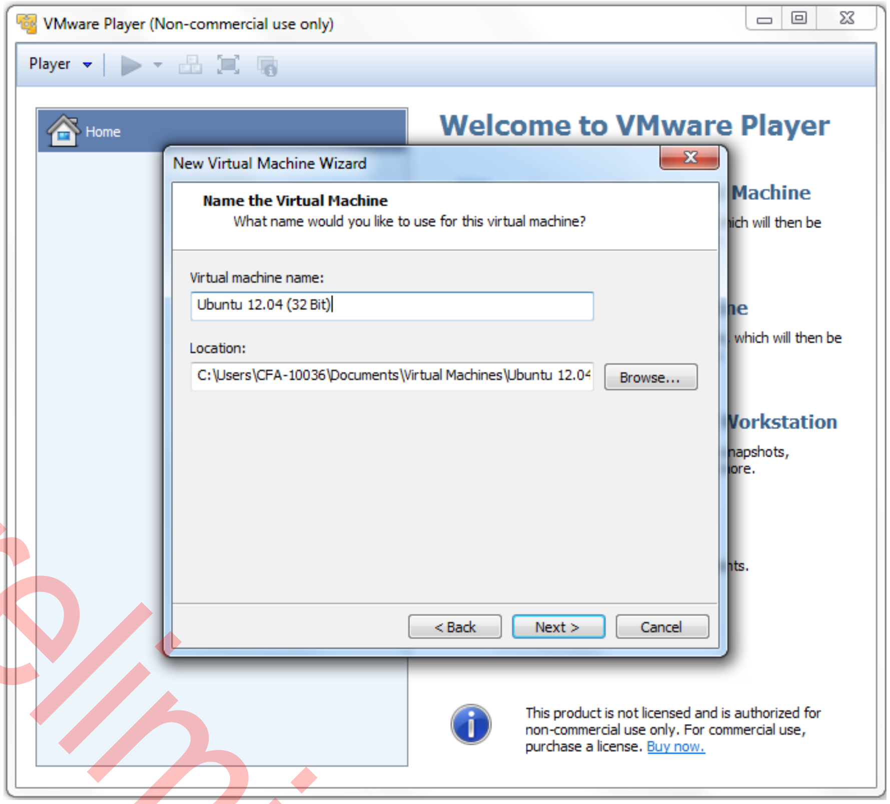
User name: cfa10036

Password: ••••••••

Confirm: ••••••••

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Enter the name you want to appear in the list of virtual machines loaded into VMware Player.


VMware Player (Non-commercial use only)
Player
Home
Welcome to VMware Player
New Virtual Machine Wizard
< Back
Next >
Cancel
Browse...

Name the Virtual Machine

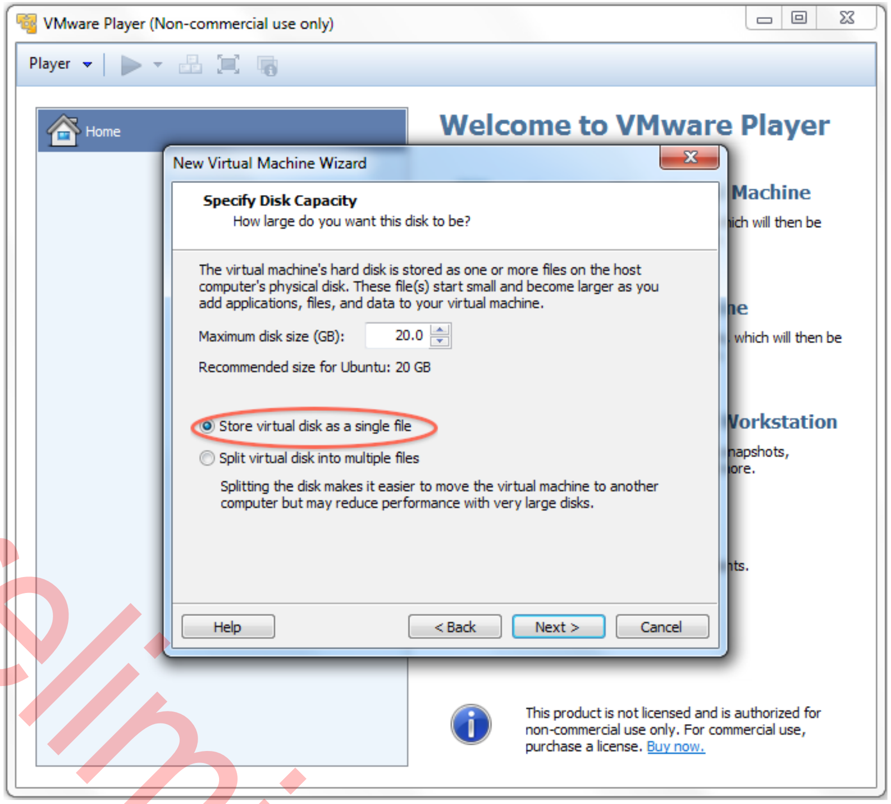
What name would you like to use for this virtual machine?

Virtual machine name: Ubuntu 12.04 (32 Bit)

Location: C:\Users\CFA-10036\Documents\Virtual Machines\Ubuntu 12.04

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Unless you plan to move the virtual machine to a different PC, select the option given.



Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): 20.0

Recommended size for Ubuntu: 20 GB

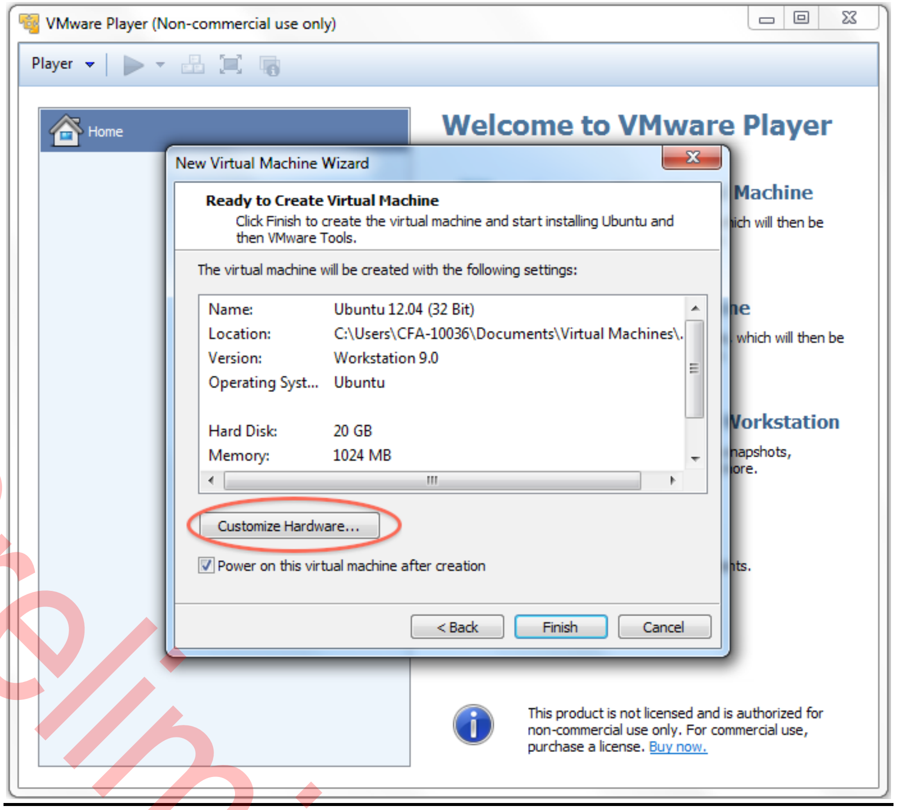
● Store virtual disk as a single file

○ Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

If you are running on a multi-core processor with a lot of memory, the default settings will suffice. You can select the Custom Hardware option if you want to assign more resources to the Ubuntu virtual machine.



Ready to Create Virtual Machine
Click Finish to create the virtual machine and start installing Ubuntu and then VMware Tools.

The virtual machine will be created with the following settings:

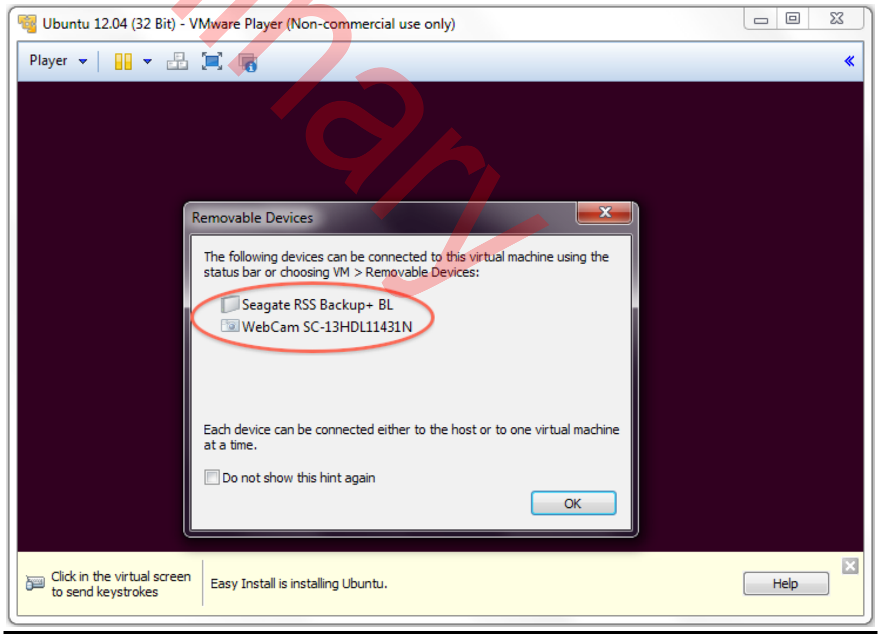
Name:	Ubuntu 12.04 (32 Bit)
Location:	C:\Users\CFA-10036\Documents\Virtual Machines\.
Version:	Workstation 9.0
Operating Syst...	Ubuntu
Hard Disk:	20 GB
Memory:	1024 MB

☑ Power on this virtual machine after creation

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Depending on your hardware, you may get messages similar to this one which lists hardware which can be dynamically swapped between the virtual machine and Windows.

When copying a Linux image to an SSD card, it will be necessary to assign the card to the virtual machine as outlined in the document describing how to build an image.


Ubuntu 12.04 (32 Bit) - VMware Player (Non-commercial use only)
Player
Removable Devices
OK
Help

The following devices can be connected to this virtual machine using the status bar or choosing VM > Removable Devices:

- Seagate RSS Backup+ BL
- WebCam SC-13HDL11431N

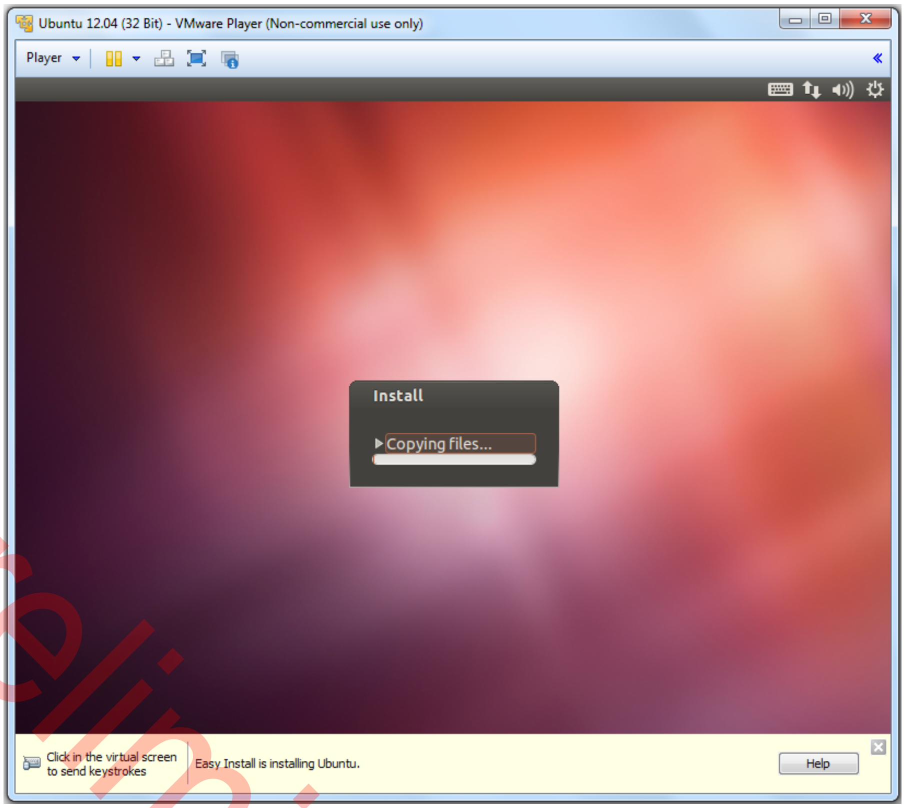
Each device can be connected either to the host or to one virtual machine at a time.

□ Do not show this hint again

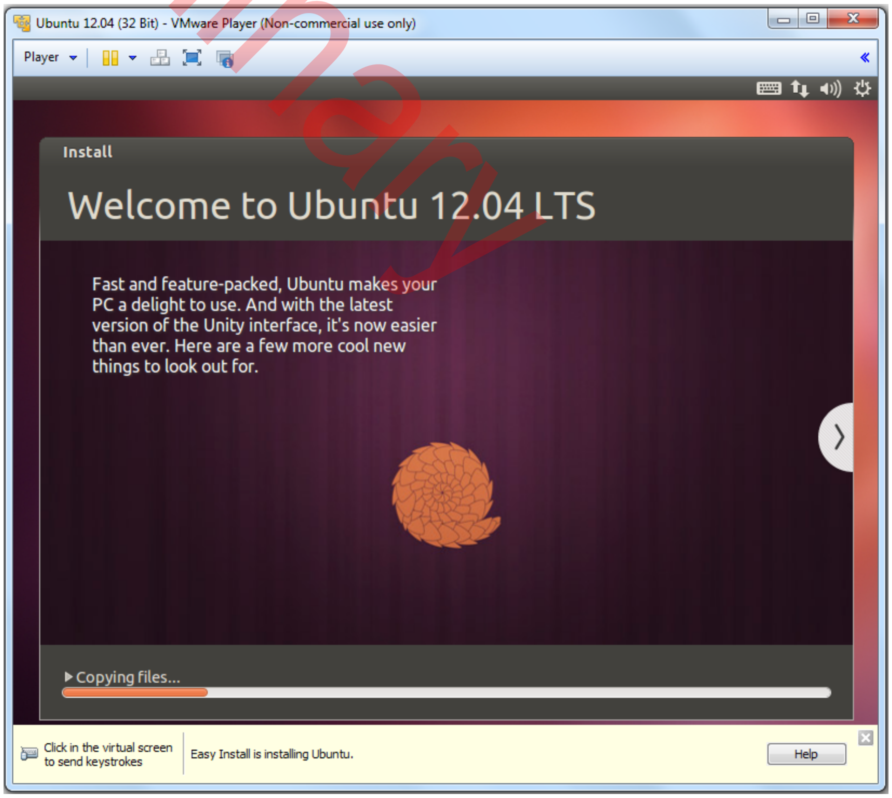
Click in the virtual screen to send keystrokes

Easy Install is installing Ubuntu.

From this point on, the Ubuntu installation is identical to installing on a Linux partition. Refer to the Ubuntu installation instructions if you need more information.

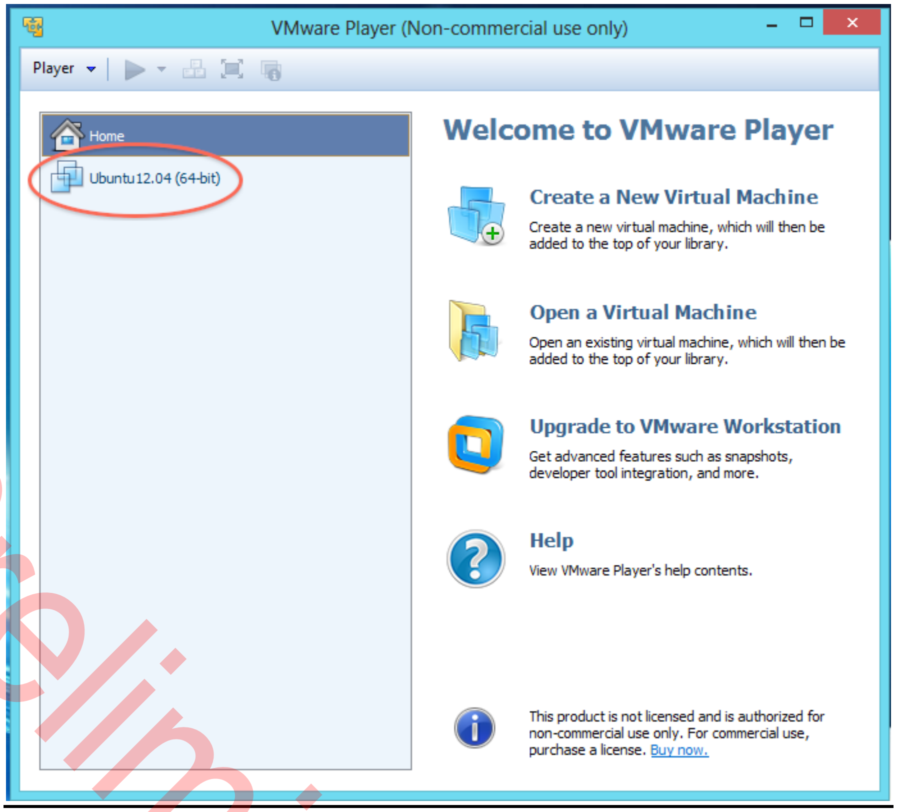

Ubuntu 12.04 (32 Bit) - VMware Player (Non-commercial use only)
Player
Install
Copying files...
Click in the virtual screen to send keystrokes
Easy Install is installing Ubuntu.
Help

.

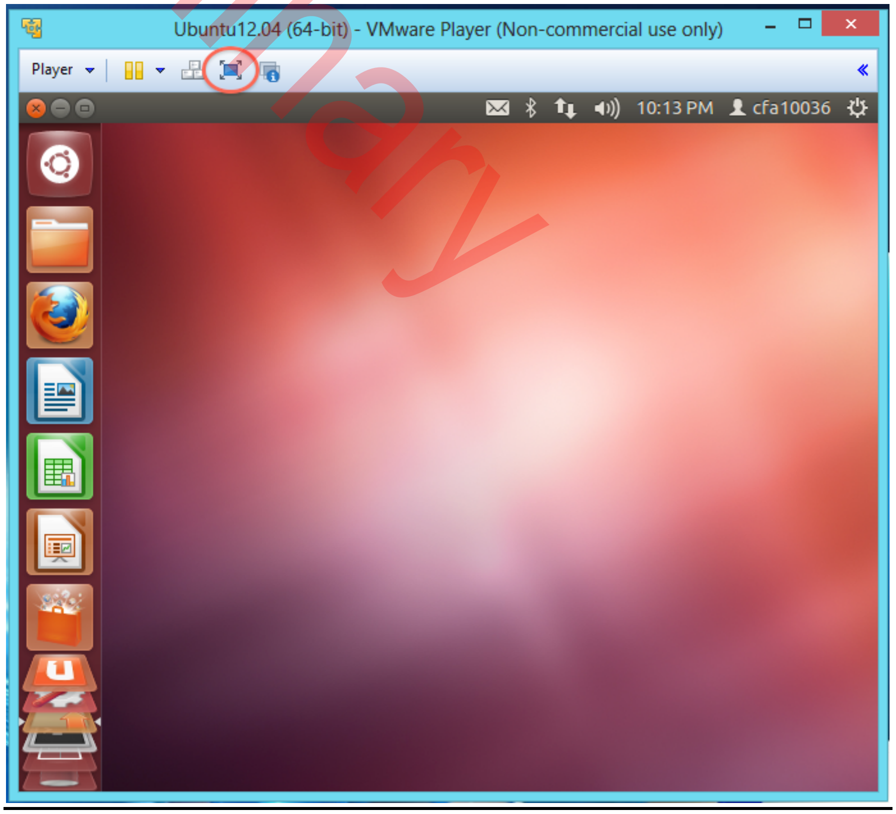

Ubuntu 12.04 (32 Bit) - VMware Player (Non-commercial use only)
Player
Install
Welcome to Ubuntu 12.04 LTS
Copying files...
Click in the virtual screen to send keystrokes
Easy Install is installing Ubuntu.
Help

Fast and feature-packed, Ubuntu makes your PC a delight to use. And with the latest version of the Unity interface, it's now easier than ever. Here are a few more cool new things to look out for.

Once Ubuntu is installed, it can be started by opening the VMware Player and selecting the Ubuntu virtual machine.


VMware Player (Non-commercial use only)
Player
Home
Ubuntu12.04 (64-bit)
Welcome to VMware Player
Create a New Virtual Machine
Create a new virtual machine, which will then be added to the top of your library.
Open a Virtual Machine
Open an existing virtual machine, which will then be added to the top of your library.
Upgrade to VMware Workstation
Get advanced features such as snapshots, developer tool integration, and more.
Help
View VMware Player's help contents.
This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Ubuntu should now be up and running as a virtual machine under Windows. If you wish to work in full screen mode while in Linux, click on the button marked with a red circle.


Ubuntu12.04 (64-bit) - VMware Player (Non-commercial use only)
Player
10:13 PM
cfa10036

APPENDIX B: HOW TO BUILD THE LINUX DISK IMAGE

THIS APPENDIX IS INCOMPLETE AND HAS NOT BEEN FULLY VALIDATED.

INTRODUCTION

In most cases, simple changes to the .dts Linux device table are all that is required to support additional hardware. But if your design requires extra functionality, you can configure the kernel to your hardware. This requires a Linux PC or a Windows 7 or 8 VM (virtual machine) running Linux.

Ubuntu 12.04

Ubuntu 12.04 is recommended, as it was tested with buildroot and Freescale's image creation scripts and works well. Due to the subtle differences between different versions of Linux, these instructions were verified to work on Ubuntu 12.04. All other versions of Linux may need some modification to make them work correctly. Ubuntu 12.04 can be downloaded from the following link: www.ubuntu.com/download.

Windows 7 or 8 with Ubuntu Version 12.04 Installed Using a Virtual Machine Manager

To install a virtual Linux operating system under Windows, you will need to install a virtualization platform. VMware player works well for this purpose and is free for noncommercial use. Another option is to use Virtualbox as the virtual machine manager. Virtualbox is a little more complicated to set up and may be confusing to some users. We tested VMware under Windows 7 and 8 and it works well. You can download VMware player from the following link: www.vmware.com/products/player.

Use VMware to install the Ubuntu 12.04 OS. See APPENDIX A: HOW TO INSTALL UBUNTU ON WINDOWS 8 USING VMWARE PLAYER (Pg. 63).

STEP 1: INSTALL ADDITIONAL UBUNTU 12.04 PACKAGES

Some packages required by buildroot are not included in the Ubuntu 12.04 image. These packages need to be installed before buildroot will run.

Use the command line in a terminal window and type in the following:

```
sudo apt-get install g++
sudo apt-get install bison
sudo apt-get install flex
sudo apt-get install gettext
sudo apt-get install ncurses-dev
sudo apt-get install texinfo
sudo apt-get install git
```

If you are using Ubuntu version later than 12.04, check to see if these packages are included.

STEP 2: DOWNLOAD BUILDROOT

Once the above packages have been installed, download buildroot

```
 git clone https://github.com/crystalfontz/cfa_10036_buildroot
```

Buildroot will be downloaded to a directory `cfa_10036_buildroot.`

NOTE: These instructions will change once all of the Crystalfontz modifications have been pushed to the master.

Enter the following commands to build an image:

THESE INSTRUCTIONS MAY CHANGE

```
cd cfa_10036_buildroot
make crystalfontz_cfa10036_defconfig
make linux-menuconfig
```

Change the options in the configuration menu to support your hardware. Then type:

```
make
```

Buildroot will start downloading and building a toolchain and all packages necessary for a Linux board support package for the CFA-10036.

STEP 3: LOAD DISK IMAGE ONTO SD CARD

Once the disk image has been built, it can be loaded onto an SD card using the following command:

```
>    sudo python board/crystalfontz/cfa-10036/mk_cfa_sd.py \
>    -e output/images/barebox-env \
>    -b output/images/cfa10036_barebox_ivt.sb \
>    -r output/images/rootfs.tar /dev/$device
```

Where device is the name of the device assigned to the SD card.

APPENDIX C: QUALITY ASSURANCE STANDARDS

INSPECTION CONDITIONS

- ❑ Environment
  - ■ Temperature: 25±5°C
  - ■ Humidity: 30~85% RH

ACCEPTANCE SAMPLING

DEFECT TYPE	AQL*
Major	≤.65%
Minor	<1.0%
* Acceptable Quality Level: maximum allowable error rate or variation from standard	

DEFECTS CLASSIFICATION

Defects are defined as:

- A *major defect* is a defect that substantially reduces usability of unit for its intended purpose.
- A *minor defect*: is a defect that is unlikely to reduce usability for its intended purpose.

ACCEPTANCE STANDARDS

#	DEFECT TYPE	ACCEPTANCE STANDARDS CRITERIA	MAJOR / MINOR
1	PCB defects	1. Oxidation or contamination on connectors.* 2. Wrong parts, missing parts, or parts not in specification.* 3. Jumpers set incorrectly. 4. Solder (if any) on bezel, LED pad, zebra pad, or screw hole pad is not smooth. **Minor if display functions correctly. Major if the display fails.*	Minor
2	Soldering defects	1. Unmelted solder paste. 2. Cold solder joints, missing solder connections, or oxidation.* 3. Solder bridges causing short circuits.* 4. Residue or solder balls. 5. Solder flux is black or brown. **Minor if display functions correctly. Major if the display fails.*	Minor

APPENDIX D: VIBRATION TEST REPORT

KeyTronicEMS
Trust. Commitment. Results.

Test: Sine & Random Vibration

Reliability Laboratory

Originator: Brent Crosby – Crystalfontz America
Test Coordinator: Larry Bettinger - lbetting@keytronic.com 509-927-5577
Test Started: April 9, 2013
Test Completed: April 10, 2013

Summary:

The following Crystalfontz America samples were submitted for operational vibration testing:

Sample Description	S/N
533 Yellow	1148533YYHD063605
533 Blue	1234533TMITD075774
633 Yellow	1037633YYH297069
633 White	1217633TFHD356000
735 Yellow	1212735TFK0002778
735 White	1212735TFK0002778
2x CFA-10036 ver. 1.0	Pilot run samples, no S/N assigned.
2x CFA-10037 ver. 1.0	Pilot run samples, no S/N assigned.

Test Conditions:

The samples were mounted to a customer's fixture plate which was bolted directly to the slip table for the X and Y-axes. For the Z-axis the fixture was bolted to the tester with a small aluminum coupling plate. The vibration testing was performed on a Ling Dynamic Systems V730 vibrator with a Data Physics SignalStar Scalar vibration control system version 2.2.923. The samples were subjected to following profiles:

- GR-63-CORE 5.4.2, Office Vibration, Alternative Test: 5-100-5 Hz at 1.0 g with a sweep rate of .25 octave/minute, 35 minutes per axis.
- MIL-STD 810F, Figure 514C-17, Random: 1 hour per axis.
- MIL-STD 810F, Figure 514C-18, Sine: 1 hour per axis.

Equipment used:

Equipment	Model	S/N	Calibration Due Date
Endevco Control Accelerometer	7221	AM67	12-03-13
Endevco Charge amplifier	2721B	ER01	12-03-13
Data Physics Vibration controller	SignalStar Scalar	74244	05-29-13

KeyTronicEMS
Trust. Commitment. Results.

Test: Sine & Random Vibration

Reliability Laboratory

Originator: Brent Crosby – Crystalfontz America
Test Coordinator: Larry Bettinger - lbetting@keytronic.com 509-927-5577
Test Started: April 9, 2013
Test Completed: April 10, 2013

Summary:

The following Crystalfontz America samples were submitted for operational vibration testing:

Sample Description	S/N
533 Yellow	1148533YYHD063605
533 Blue	1234533TMITD075774
633 Yellow	1037633YYH297069
633 White	1217633TFHD356000
735 Yellow	1212735TFK0002778
735 White	1212735TFK0002778
2x CFA-10036 ver. 1.0	Pilot run samples, no S/N assigned.
2x CFA-10037 ver. 1.0	Pilot run samples, no S/N assigned.

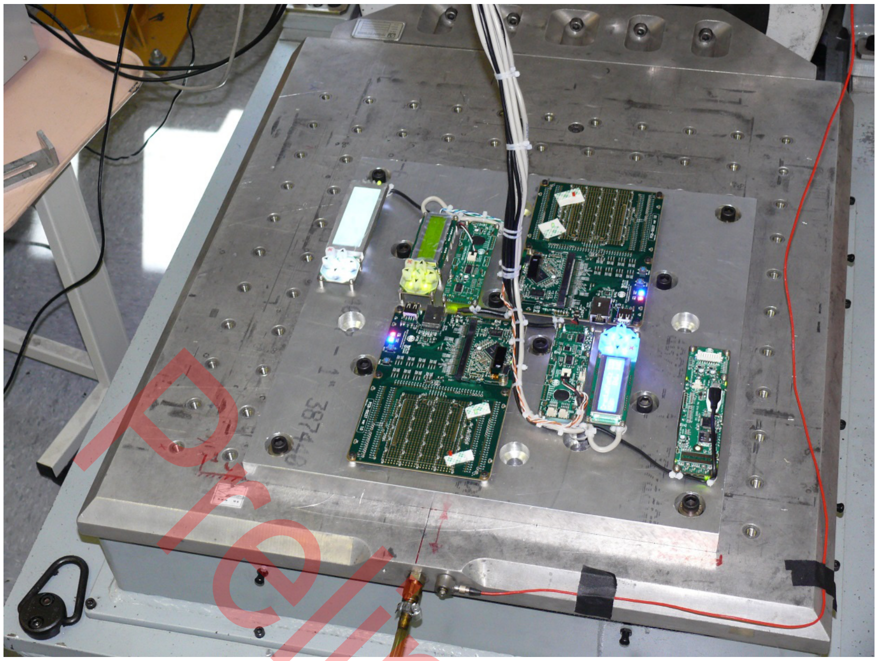
Test Conditions:

The samples were mounted to a customer's fixture plate which was bolted directly to the slip table for the X and Y-axes. For the Z-axis the fixture was bolted to the tester with a small aluminum coupling plate. The vibration testing was performed on a Ling Dynamic Systems V730 vibrator with a Data Physics SignalStar Scalar vibration control system version 2.2.923. The samples were subjected to following profiles:

- GR-63-CORE 5.4.2, Office Vibration, Alternative Test: 5-100-5 Hz at 1.0 g with a sweep rate of .25 octave/minute, 35 minutes per axis.
- MIL-STD 810F, Figure 514C-17, Random: 1 hour per axis.
- MIL-STD 810F, Figure 514C-18, Sine: 1 hour per axis.

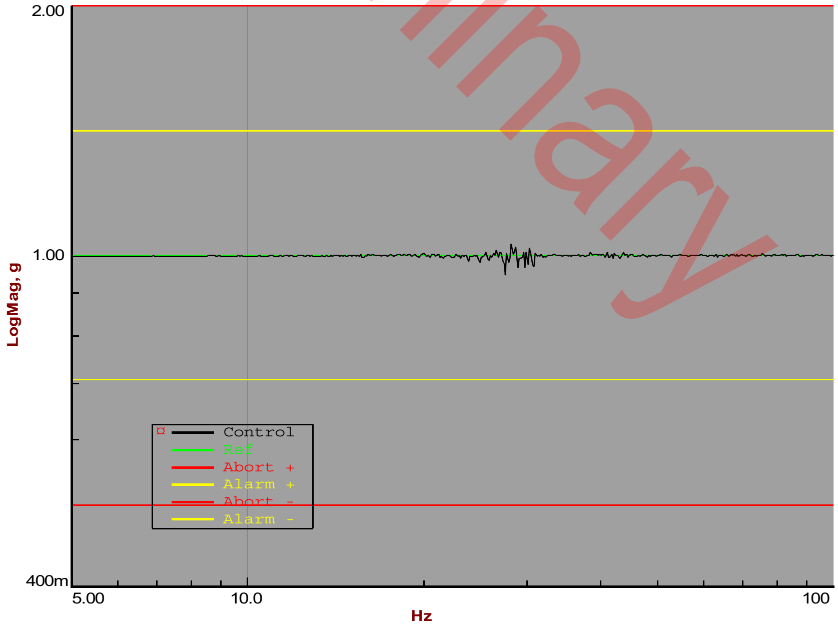
Equipment used:

Equipment	Model	S/N	Calibration Due Date
Endevco Control Accelerometer	7221	AM67	12-03-13
Endevco Charge amplifier	2721B	ER01	12-03-13
Data Physics Vibration controller	SignalStar Scalar	74244	05-29-13



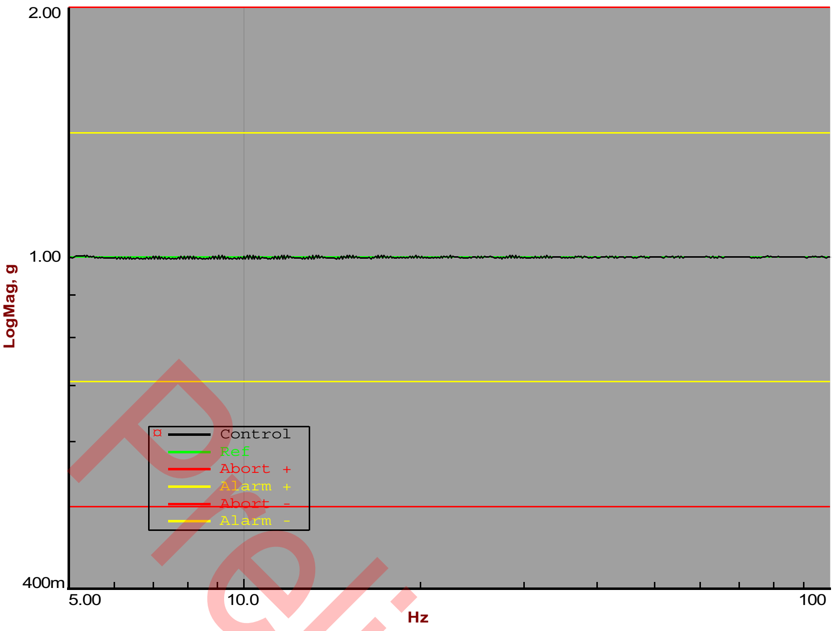
X-axis

Control accelerometer vibration level graphs:

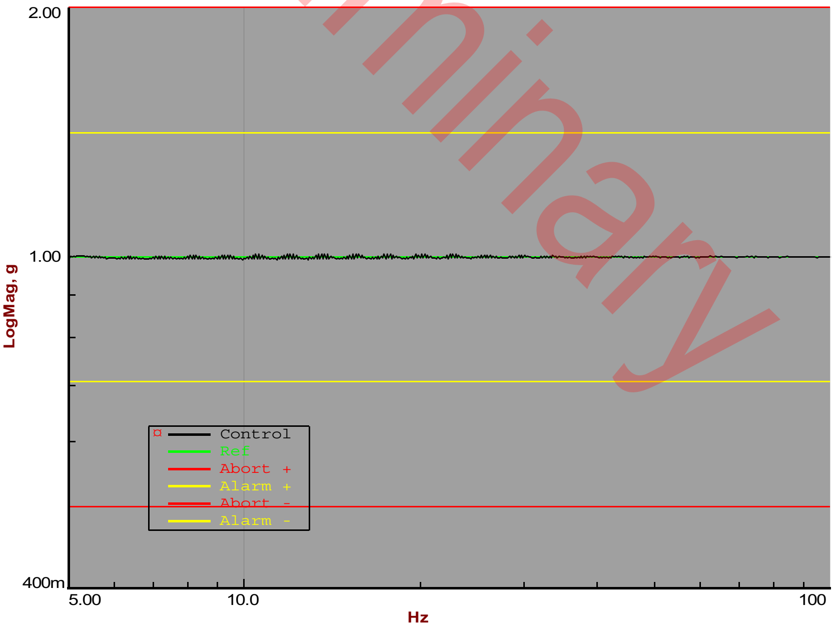


Z-axis (GR-63-CORE 5.4.2, Office Vibration, Alternative Test)

Test Report Number: CRYSTALFONTZ002 Page 3 of 7

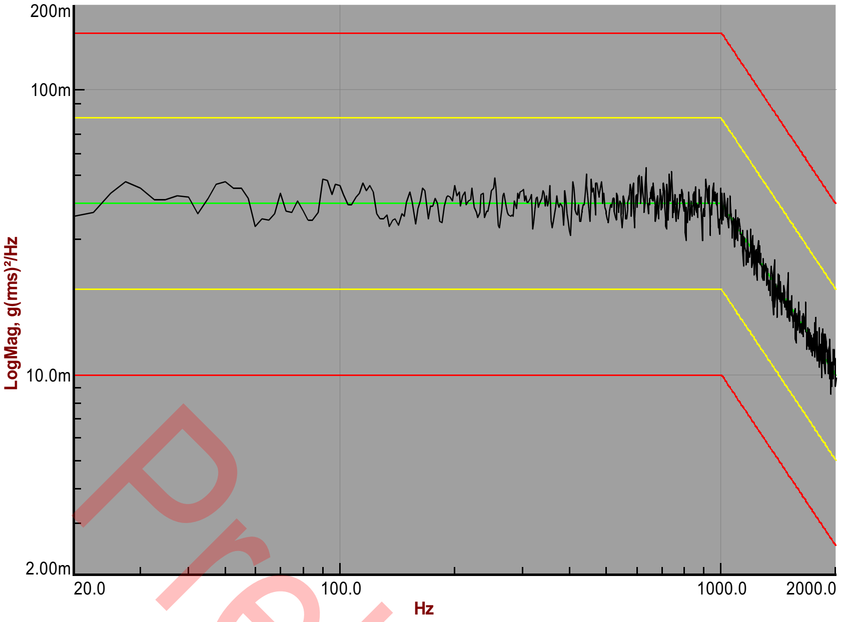


Y-axis (GR-63-CORE 5.4.2, Office Vibration, Alternative Test)

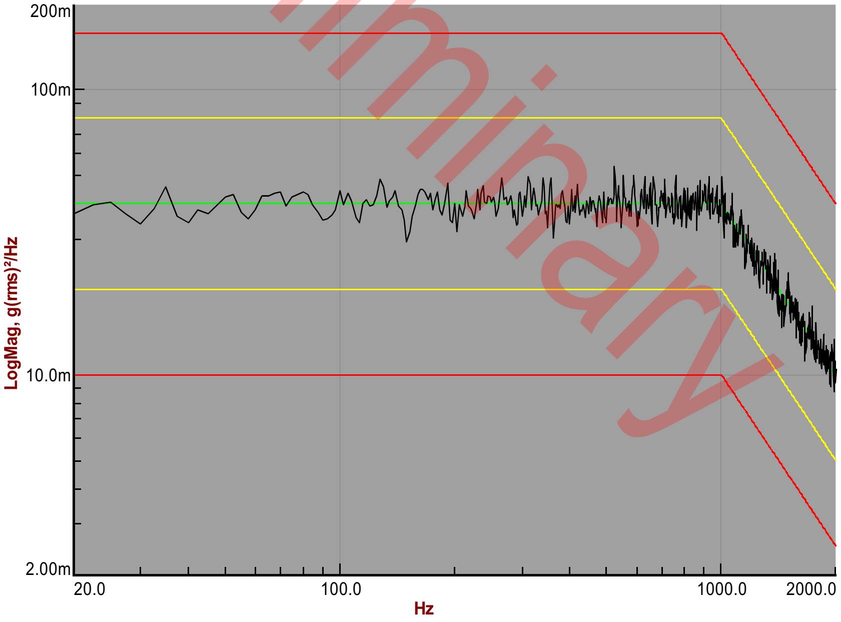


X-axis (GR-63-CORE 5.4.2, Office Vibration, Alternative Test)

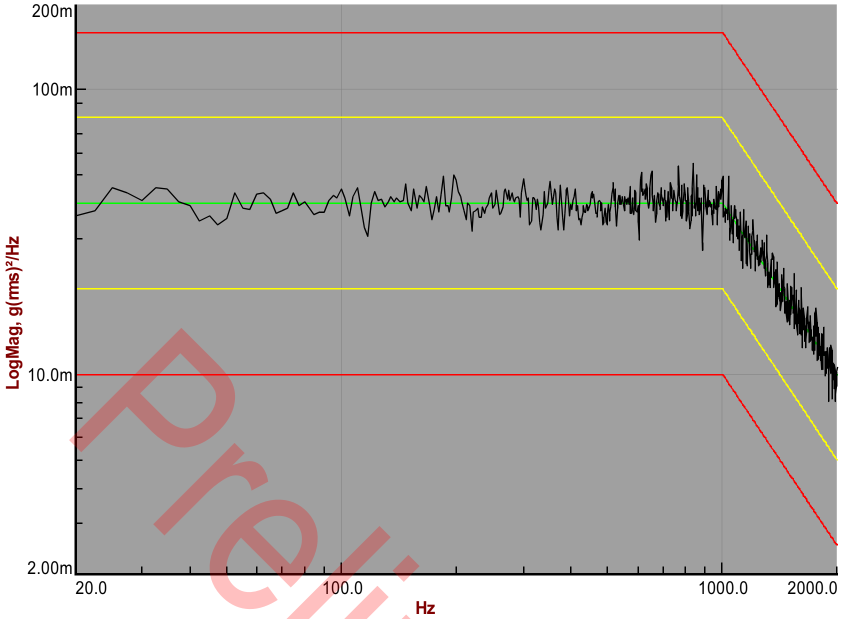
.



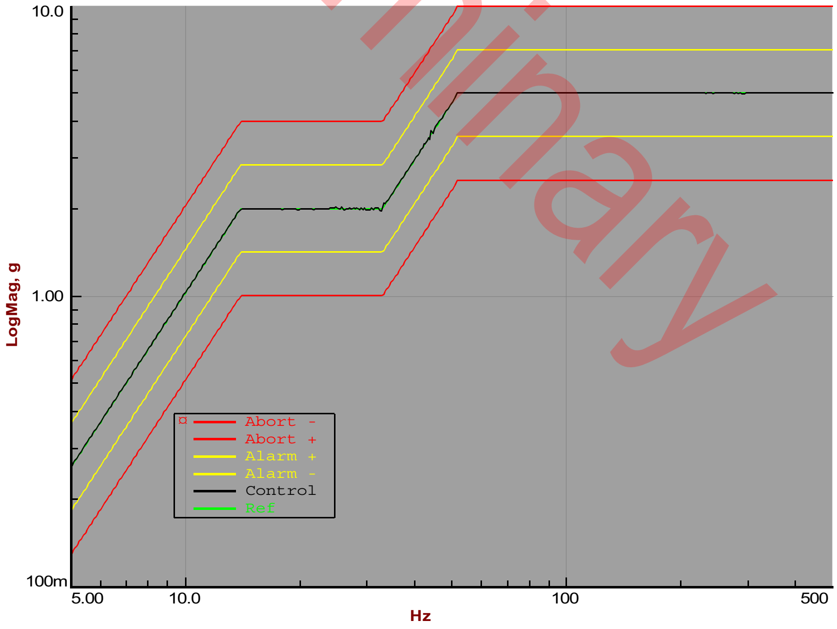
X-axis (MIL-STD 810F, Figure 514C-17, Random)



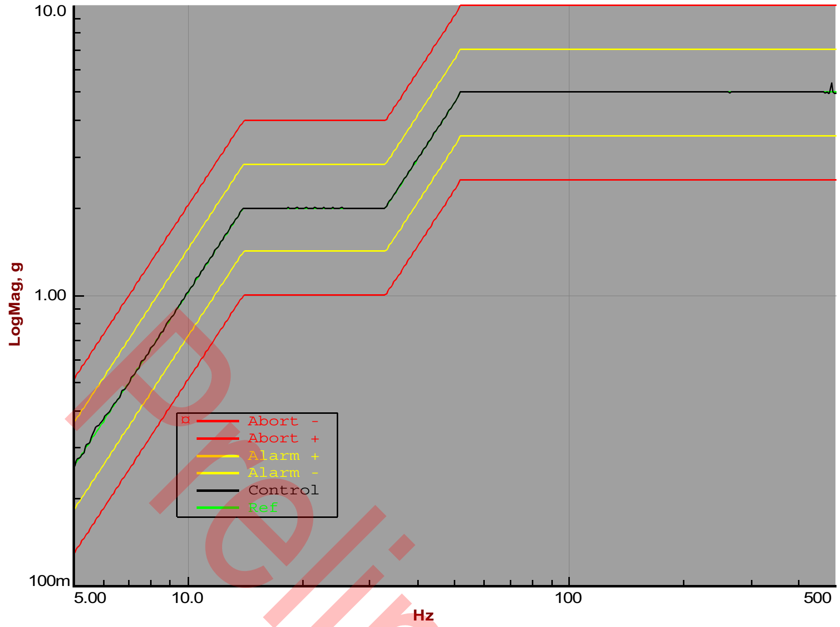
Y-axis (MIL-STD 810F, Figure 514C-17, Random)



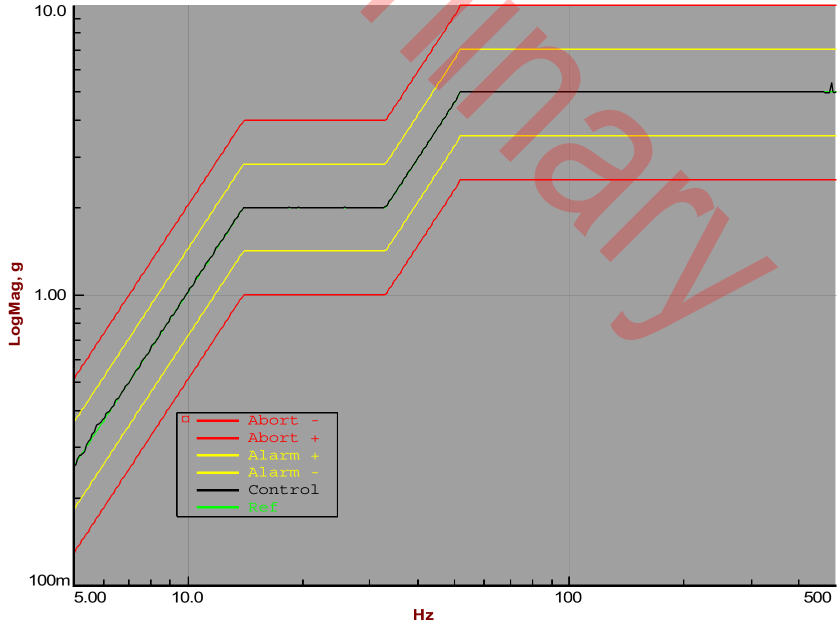
Z-axis (MIL-STD 810F, Figure 514C-17, Random)



Z-axis (MIL-STD 810F, Figure 514C-18, Sine)



Y-axis (MIL-STD 810F, Figure 514C-18, Sine)



X-axis (MIL-STD 810F, Figure 514C-18, Sine)