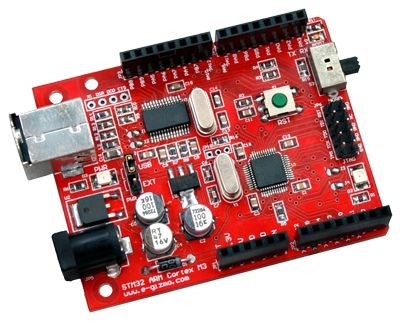
**STM32: MCU Board**

**(STM32F100C8)**

**Hardware Manual Rev 1r0**

*Develop projects with a 32-bit processing power at 8-bit cost. e-Gizmo’s STM32 platform kit follows the Arduino style pin layout that allows you to use it with existing Gizduino shields and training modules.*

****

Features & Specifications

* Microcontroller: STM32F100C8
* User Interface: USB Port, DC Jack, Reset Button, JTAG Header, Boot switch, Shield Connection Port
* Power Input:
  + External: 8V-12V
  + USB: 5V
* DC Power Output: 3.3V
* PCB Size: 2.7 x 2.1 inch
* 24 MHz operation, 30 DMIPS, 8 MHz on-board crystal
* Communication Interfaces:
  + 2 I2C interfaces
  + 3 USARTS
  + 2 SPIs
* 37 GPIOs
* 10-channel 12-bit ADC

The STM32 platform kit is a microcontroller board based on the STM32F100C8 - a 32-bit ARM Cortex-M3 microcontroller operating at 24 MHz frequency with 64K Flash Memory and 8K SRAM. The board has 14 general purpose input/output (GPIO) pins and 6-channel 12-bit analog-to-digital converter (ADC) that can be remapped through firmware to utilize the chip's 37 GPIOs and 10-channel ADC. The board also has USB to UART Bridge for ISP programming and user defined UART to USB transactions, a power jack, a JTAG header and a reset button. Simply connect the board to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The STM32 kit is an open-source computing platform that you can use to implement your designed programs. The board can be programmed using command-line programming, full-feature Integrated Development Environment (IDE) such as Kiel MDK-ARM and IAR workbench or light-weight IDEs specifically developed for starters such as the one available for download at:

<http://projectproto.blogspot.com/2013/04/stm32-gcc-arm-ide.html>

Major Components

<Insert Board Diagram Featuring Main Components>

Parts & Pins Description

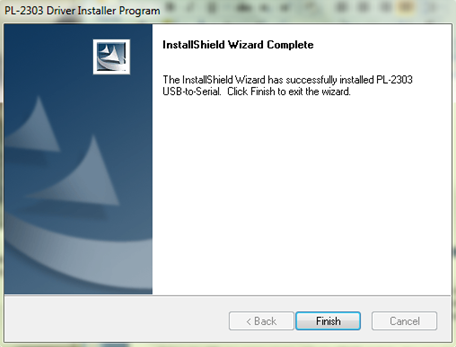
<Insert Tables of PINs, Power Jumper Diagram and Boot Switch>

Programming Setup

<Insert Diagram For Connection>

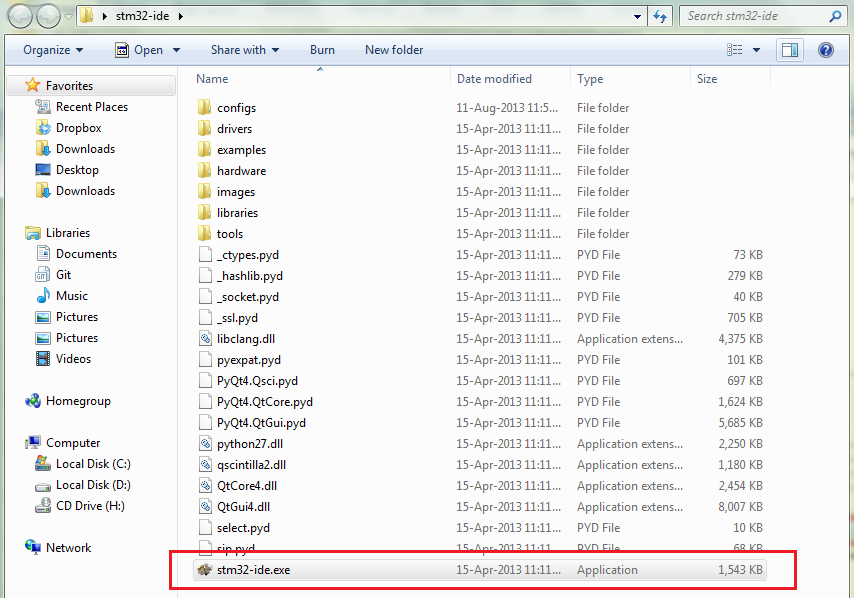
To begin, acquire the software and driver required to run STM32. The STM32-to-PC connection requires the PL-2303 USB-to-Serial driver (<http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41>). For the IDE, a lightweight version can be downloaded from <http://projectproto.blogspot.com/2013/04/stm32-gcc-arm-ide.html>. (Latest version as of this writing is svn-r158. Size: 38.7MB). You may also need 7zip (<http://www.7-zip.org/download.html>) to extract the contents of the IDE archive.

Run the ‘PL2303\_Prolific\_DriverInstaller’ executable file to initiate the installation process. Simply follow the steps indicated in the window prompt. Wait until installation is done. After installing the driver, we can now proceed with setting up the STM32 IDE.

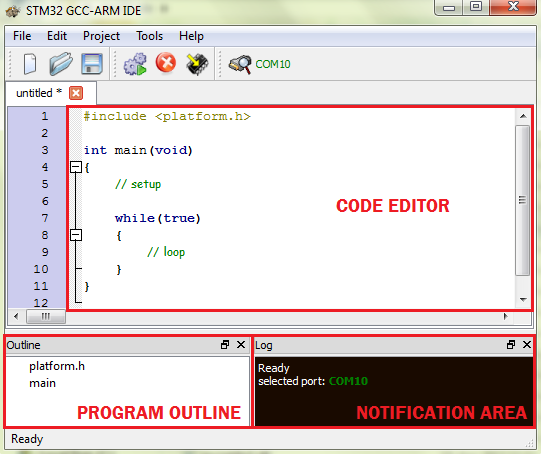


Extract the contents of the STM32 IDE using 7zip. Once extracted, we strictly recommend that the user does not make any changes to the internal structures and directories within the folder. The program is written to find its needed data according to exactly how the files are initially named and structured. Thus, any changes in the content would lead to errors.

The STM32 development environment is where all the programming codes and instructions are made and uploaded into the STM32 Platform Kit. To use this, open the folder you extracted (‘stm32-ide’ by default). You will see several folders and files contained within the IDE archive. Search for ‘stm32-ide.exe’ and run the program.

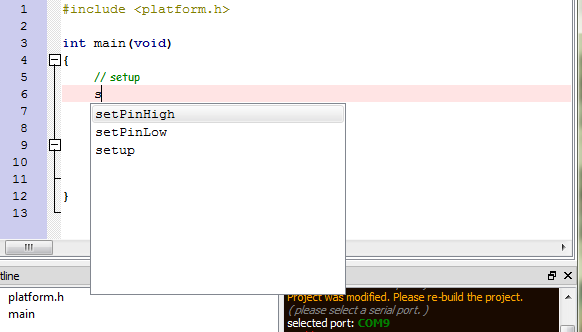


You shall see the IDE Window.

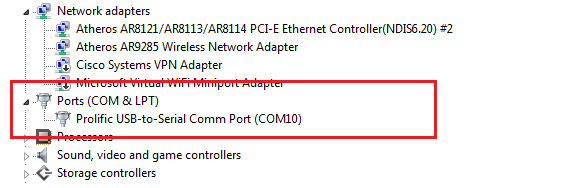


The STM32 IDE contains the ‘code editor’ – an area for writing and editing code. Notice that the ‘code editor’ already contains the main block and the required include file. This lessens potential error due to accidentally forgetting the required components in running the program. The ‘program outline’ lists all the included libraries, functions, variables, etc., within the program. This lets the user have a quick look on the components written when the code is already long. The ‘notification area’ is a section that displays log of the events that are occurring within the program such as the status of the upload process, re-building requirement, compilation error and status of the COM Port.

Within the IDE are several features included for a better user-interface. The IDE displays the line number alongside the ‘code editor’ for easier code reference and debugging. It also has ‘collapsible code block’ to let the user hide blocks of the program to see the general code structure, or show these blocks to review code details. There is also an ‘auto-complete’ feature that lists all functions that starts with the first character written to save the user from having to memorize the functions supported in the IDE. Typing the first three characters of a function or variable will automatically display candidates. Or, to manually access this feature, simply type-in the first character of the command and hit CTRL+SPACE BAR.

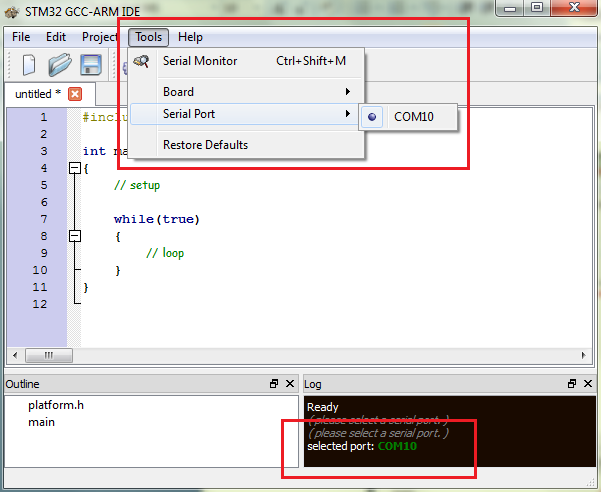


Before the IDE can communicate with the STM32 module, the user must first set the number of the COM port assigned to the USB cable. To identify the COM Port, go to the Device Manager and expand the ‘Ports (COM & LPT)’ category. Take note of the number attached to the item labeled ‘Prolific USB-to-Serial Comm Port’.

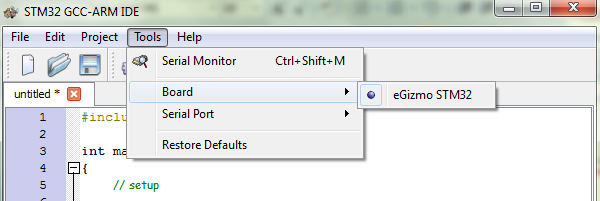


As an example, the figure above shows that our board is attached to COM Port 10. Note that if the ‘Prolific USB-to-Serial Comm Port’ item does not appear, it is probable that the PL2303 Driver did not install correctly. In such case, you may need to reinstall the driver or reset your computer.

Set the port in the IDE by going to ‘Tools -> Serial Port -> COM’. The selected port should then appear in the Notification Area.



The IDE is automatically configured to use the STM32 Platform Kit.

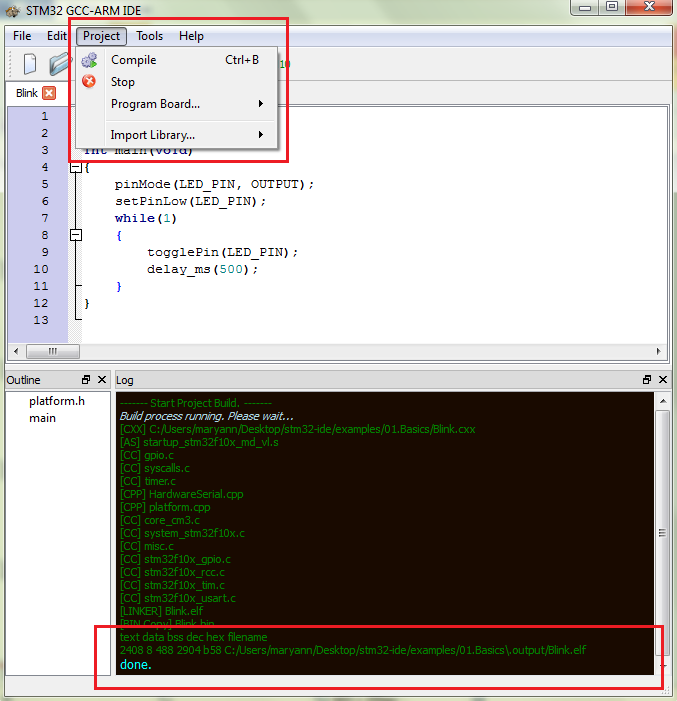


Now, we are done with our setup for programming. The next section will introduce us to our first STM32 program and will guide us on uploading our codes to the platform kit.

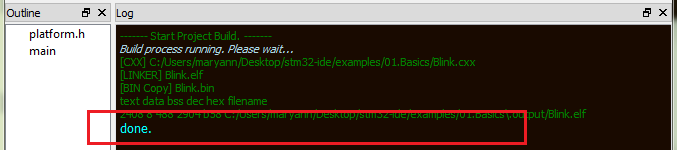
Uploading and Running a Program

The STM32 IDE contains built-in examples to help the users get started. To access these examples, go to ‘File -> Examples’ and select a program you would like to use. In our case, let’s use the ‘Blink’ example.

Compile the program by going to ‘Project -> Compile’ or hitting ‘CTRL+B’.

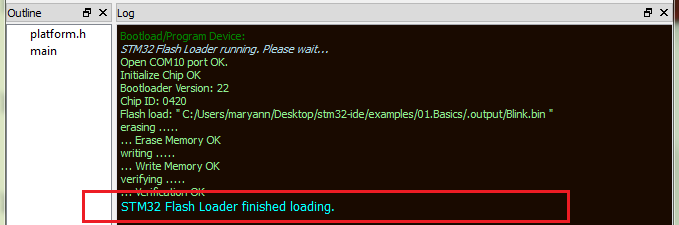


The ‘notification area’ will show the status of the compilation process. You will get a confirmation if the process is successful.



Before uploading a program, turn the position of the slider switch (located near the JTAG header) in the board to ‘Boot’ mode. If you cannot find this switch, please refer to the ‘Major Components’ Section.

Press the reset button in the board. Now, go to ‘Project -> Program Board’ and select ‘Load’ or you can hit ‘CTRL+R’. The status of the upload process will be shown on the ‘notification area’.



To see the program run, turn the position of the slider switch to ‘Norm’ mode and press the reset button. You should see the on-board LED blinking.