Embedded System Design Techniques™

Transitioning from C to C++

Class 3: Transitioning to C++

October 11th, 2017
Jacob Beningo

Presented by:

DesignNews

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Course Overview

Topics:

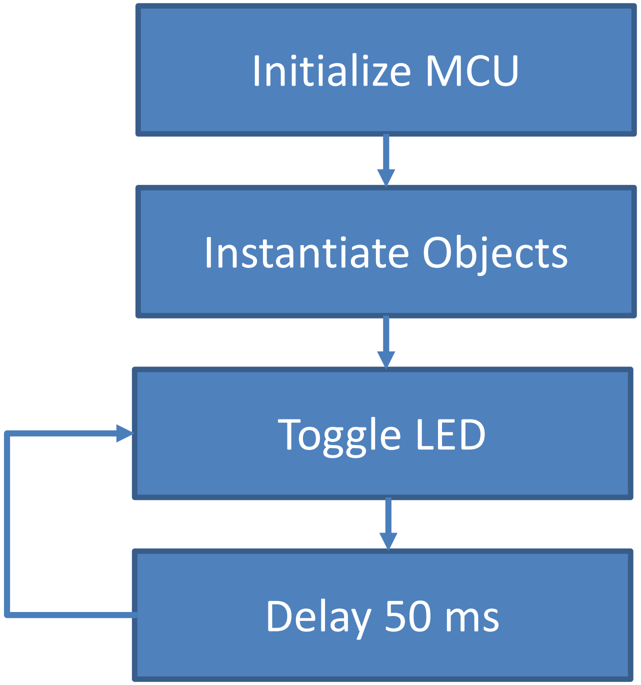
- C++ Fundamentals
- Designing a C++ Application
- **Beginning the Transition**
- Real-Time C++
- Getting into the Bits and the Bytes

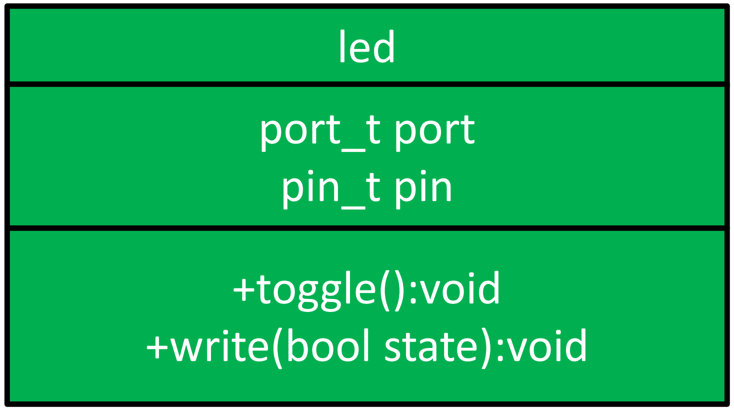
Session Overview

- The LED Blinky Program Design
- Toolchain Setup
- The LED Class
- The Blinky LED Program
- C versus C++
- Transitioning to C++
- Useful Resources

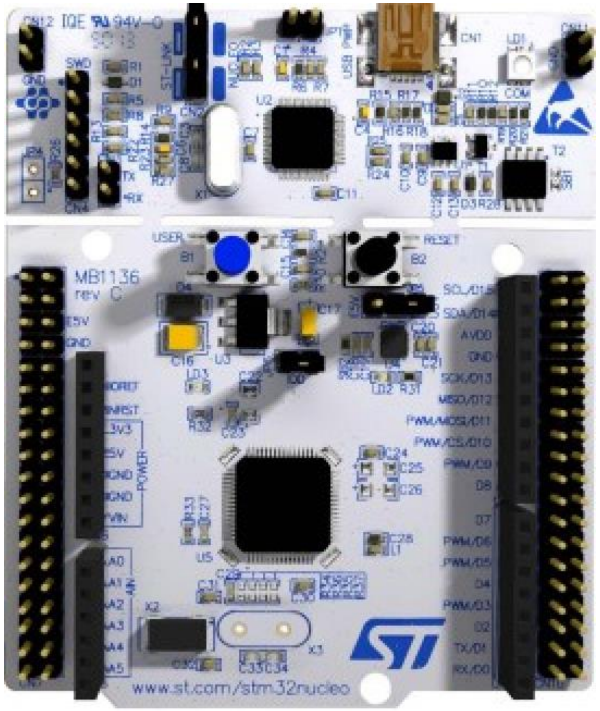
Presented by:

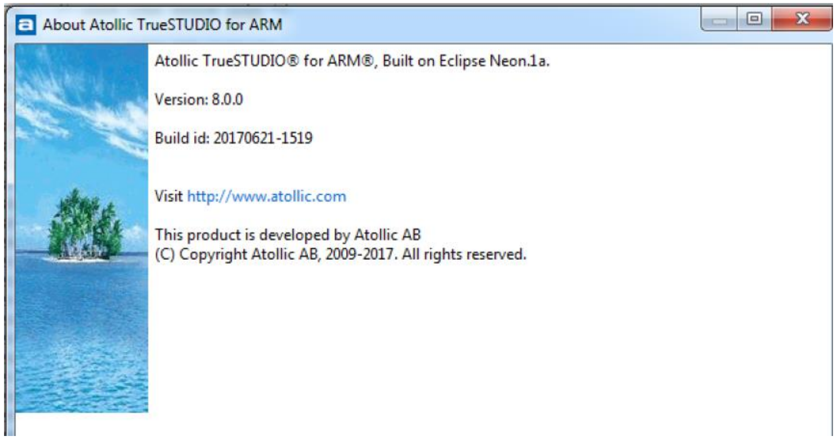
The LED Blinky Program Design





Toolchain Setup





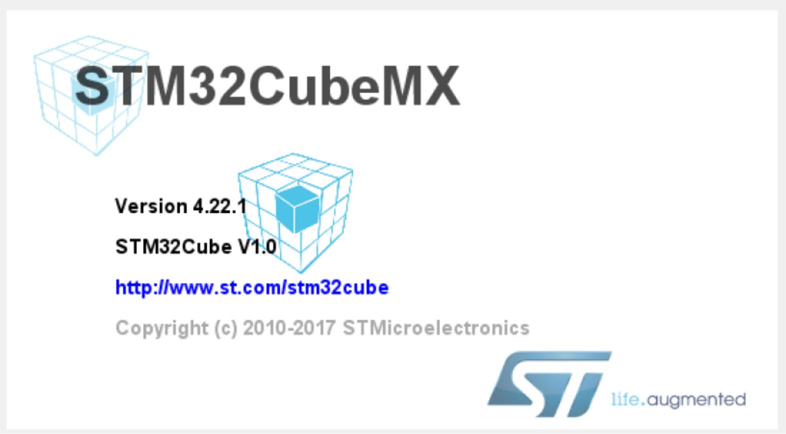
Atollic TrueSTUDIO® for ARM®, Built on Eclipse Neon.1a.

Version: 8.0.0

Build id: 20170621-1519

Visit http://www.atollic.com

This product is developed by Atollic AB
(C) Copyright Atollic AB, 2009-2017. All rights reserved.



Version 4.22.1

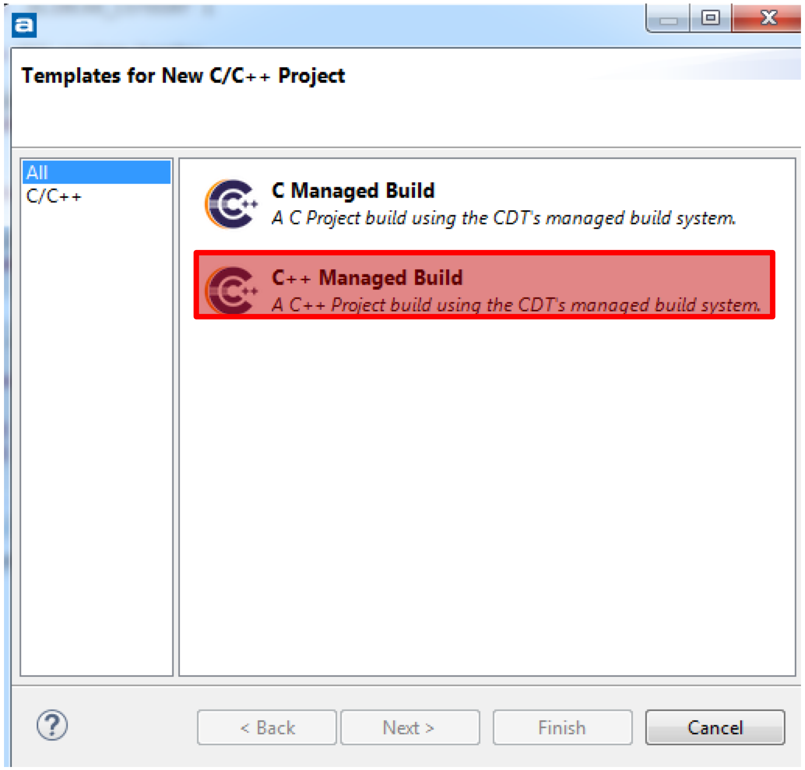
STM32Cube V1.0

http://www.st.com/stm32cube

Copyright (c) 2010-2017 STMicroelectronics

Toolchain Setup

1



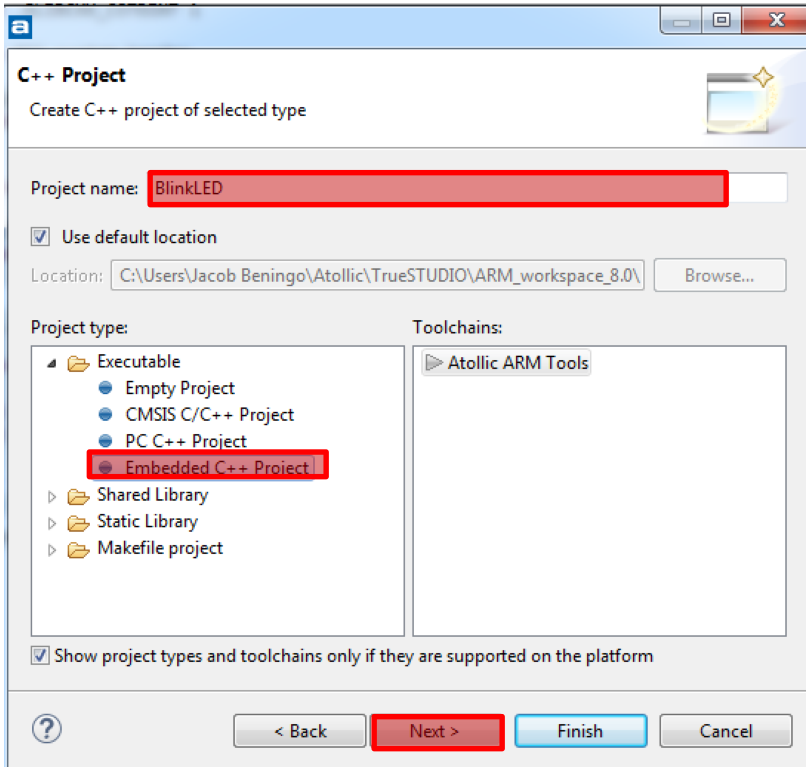
Templates for New C/C++ Project

- All
- C/C++

C Managed Build
A C Project build using the CDT's managed build system.

C++ Managed Build
A C++ Project build using the CDT's managed build system.

2



C++ Project

Create C++ project of selected type

Project name: BlinkLED

☑ Use default location

Location: C:\Users\Jacob Beningo\Atollic\TrueSTUDIO\ARM_workspace_8.0\

Project type:

- Executable
  - Empty Project
  - CMSIS C/C++ Project
  - PC C++ Project
  - Embedded C++ Project
- Shared Library
- Static Library
- Makefile project

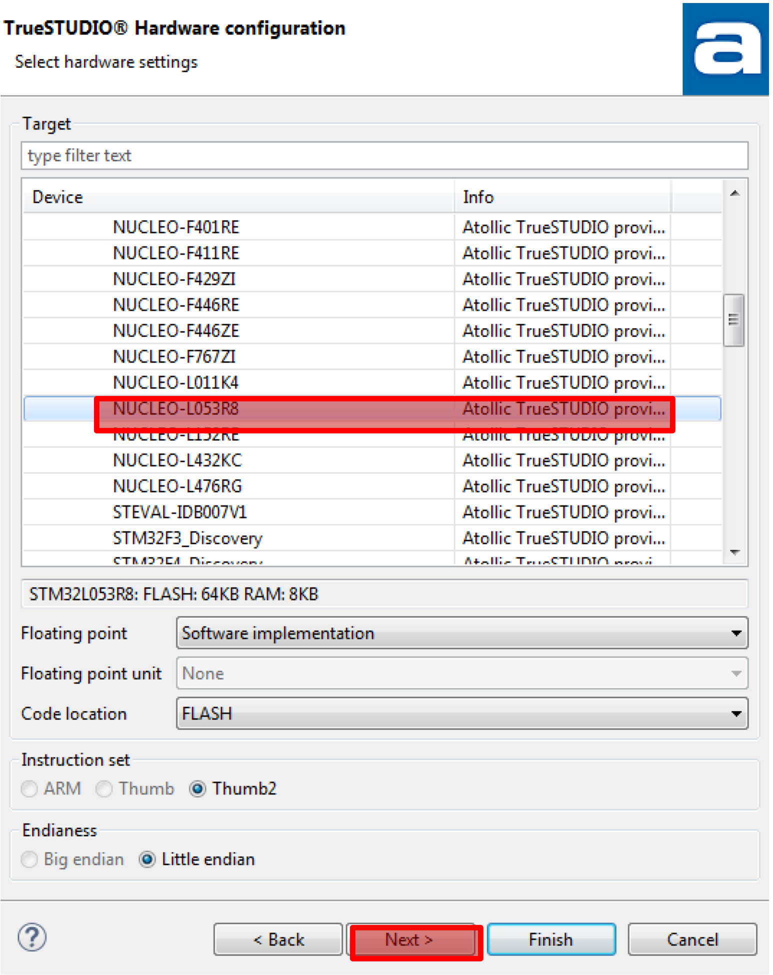
Toolchains:

- Atollic ARM Tools

☑ Show project types and toolchains only if they are supported on the platform

Toolchain Setup

3



TrueSTUDIO® Hardware configuration

Select hardware settings

Target

type filter text

Device	Info
NUCLEO-F401RE	Atollic TrueSTUDIO provi...
NUCLEO-F411RE	Atollic TrueSTUDIO provi...
NUCLEO-F429ZI	Atollic TrueSTUDIO provi...
NUCLEO-F446RE	Atollic TrueSTUDIO provi...
NUCLEO-F446ZE	Atollic TrueSTUDIO provi...
NUCLEO-F767ZI	Atollic TrueSTUDIO provi...
NUCLEO-L011K4	Atollic TrueSTUDIO provi...
NUCLEO-L053R8	Atollic TrueSTUDIO provi...
NUCLEO-L152RE	Atollic TrueSTUDIO provi...
NUCLEO-L432KC	Atollic TrueSTUDIO provi...
NUCLEO-L476RG	Atollic TrueSTUDIO provi...
STEVAL-IDB007V1	Atollic TrueSTUDIO provi...
STM32F3_Discovery	Atollic TrueSTUDIO provi...
STM32F4_Discovery	Atollic TrueSTUDIO provi...

STM32L053R8: FLASH: 64KB RAM: 8KB

Floating point: Software implementation

Floating point unit: None

Code location: FLASH

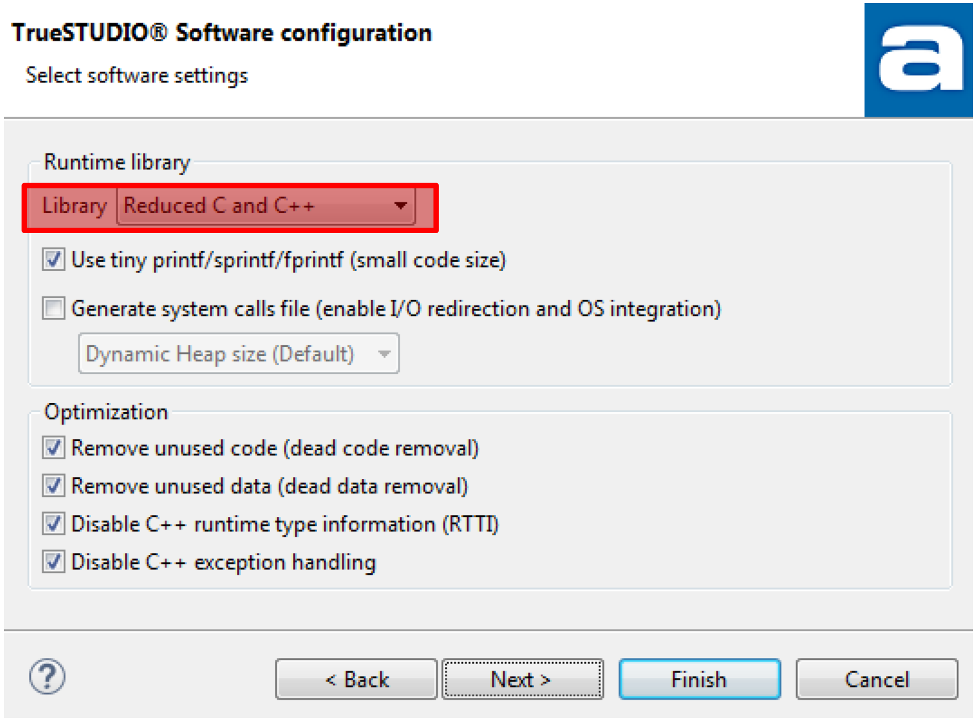
Instruction set

○ ARM ○ Thumb ● Thumb2

Endianess

○ Big endian ● Little endian

4



TrueSTUDIO® Software configuration

Select software settings

Runtime library

Library: Reduced C and C++

☑ Use tiny printf/sprintf/fprintf (small code size)

☐ Generate system calls file (enable I/O redirection and OS integration)

Dynamic Heap size (Default)

Optimization

☑ Remove unused code (dead code removal)

☑ Remove unused data (dead data removal)

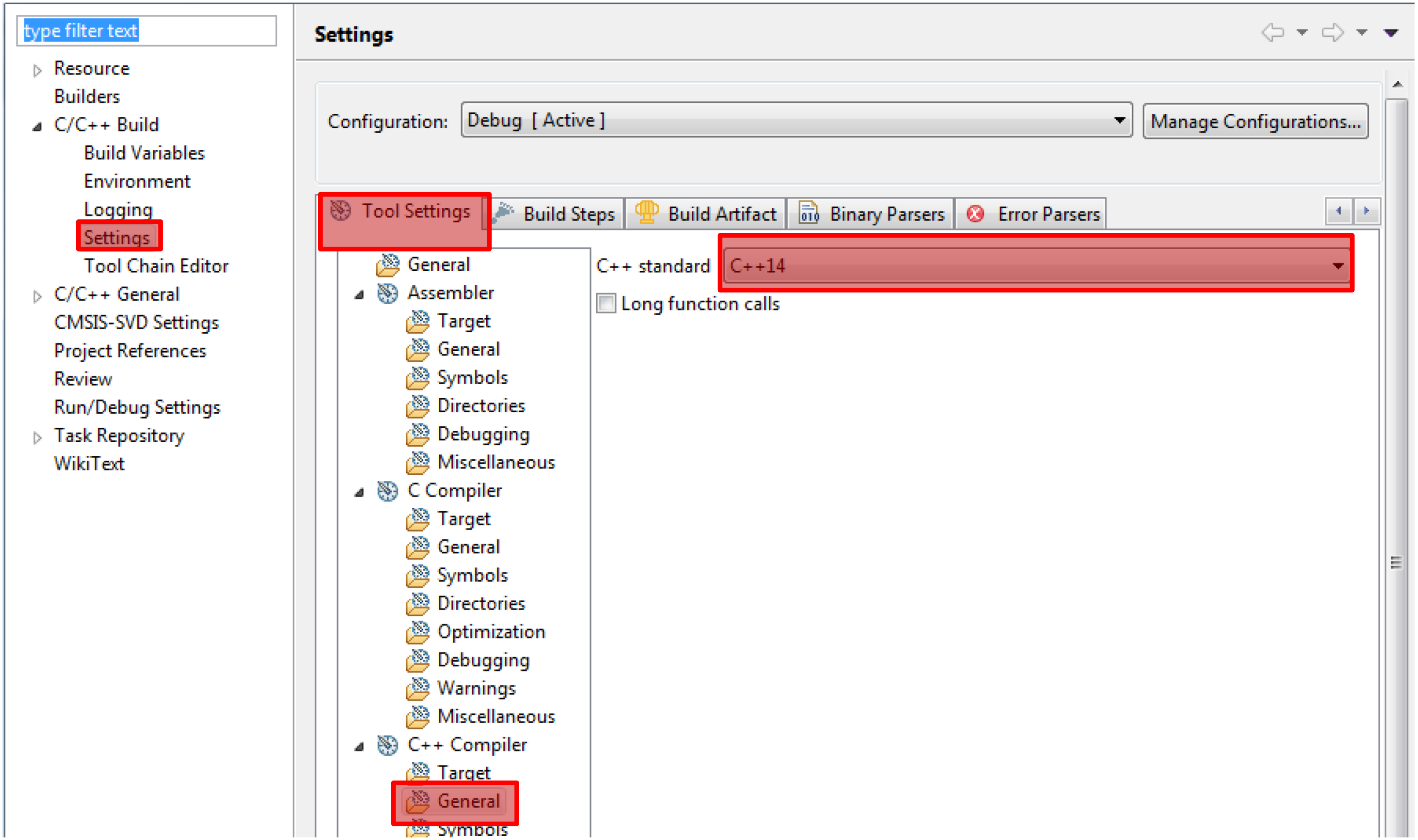
☑ Disable C++ runtime type information (RTTI)

☑ Disable C++ exception handling

Presented by:

Toolchain Setup

5



type filter text

- Resource
- Builders
- C/C++ Build
  - Build Variables
  - Environment
  - Logging
  - Settings
  - Tool Chain Editor
- C/C++ General
- CMSIS-SVD Settings
- Project References
- Review
- Run/Debug Settings
- Task Repository
- WikiText

Settings

Configuration: Debug [Active]

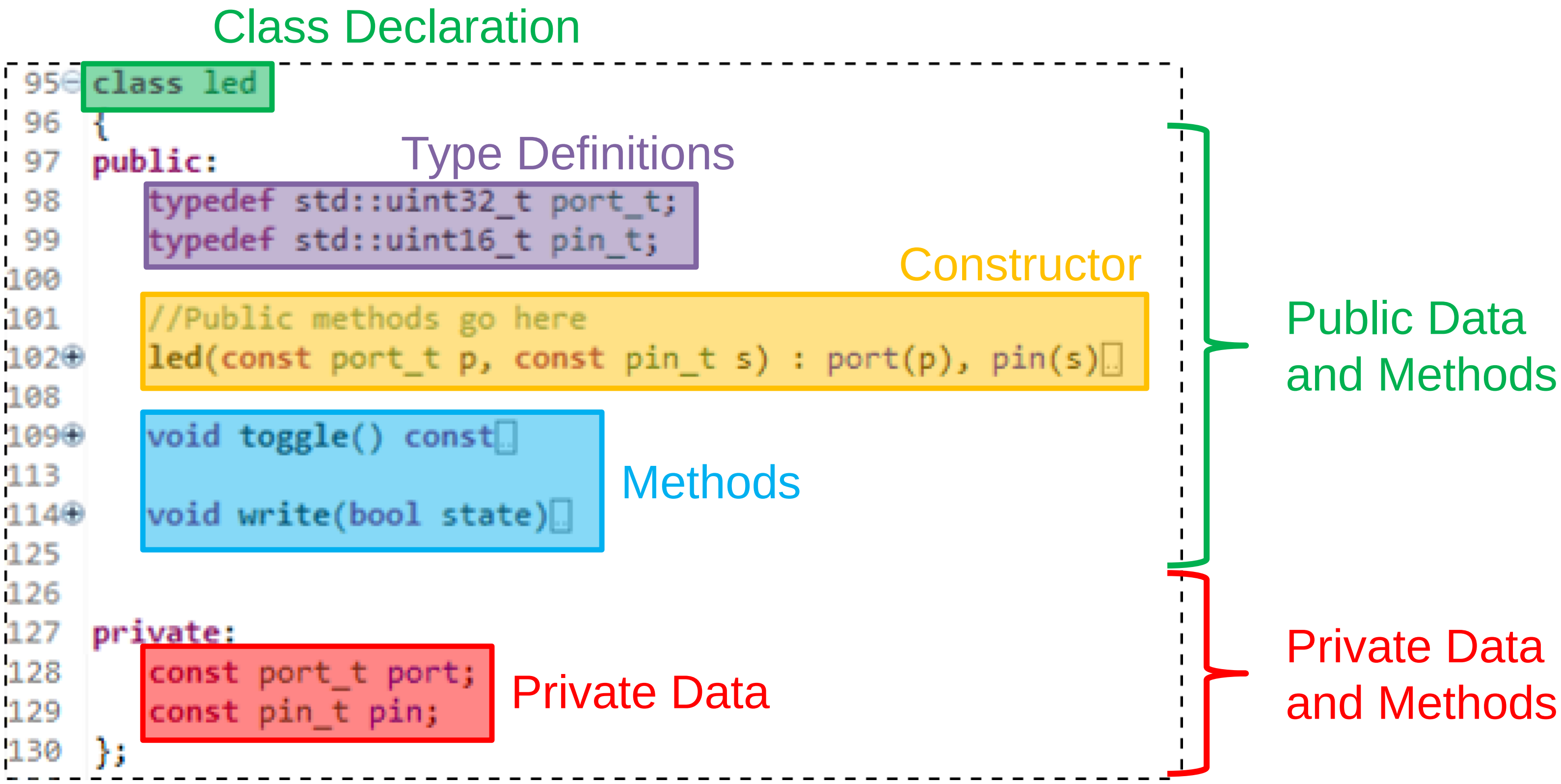
- General
- Assembler
  - Target
  - General
  - Symbols
  - Directories
  - Debugging
  - Miscellaneous
- C Compiler
  - Target
  - General
  - Symbols
  - Directories
  - Optimization
  - Debugging
  - Warnings
  - Miscellaneous
- C++ Compiler
  - Target
  - General
  - Symbols

C++ standard: C++14

□ Long function calls

Presented by:

The LED Class



Class Declaration

```
class led
{
public:
    typedef std::uint32_t port_t;
    typedef std::uint16_t pin_t;

    //Public methods go here
    led(const port_t p, const pin_t s) : port(p), pin(s)□

    void toggle() const□

    void write(bool state)□


private:
    const port_t port;
    const pin_t pin;
};
```

Type Definitions

Constructor

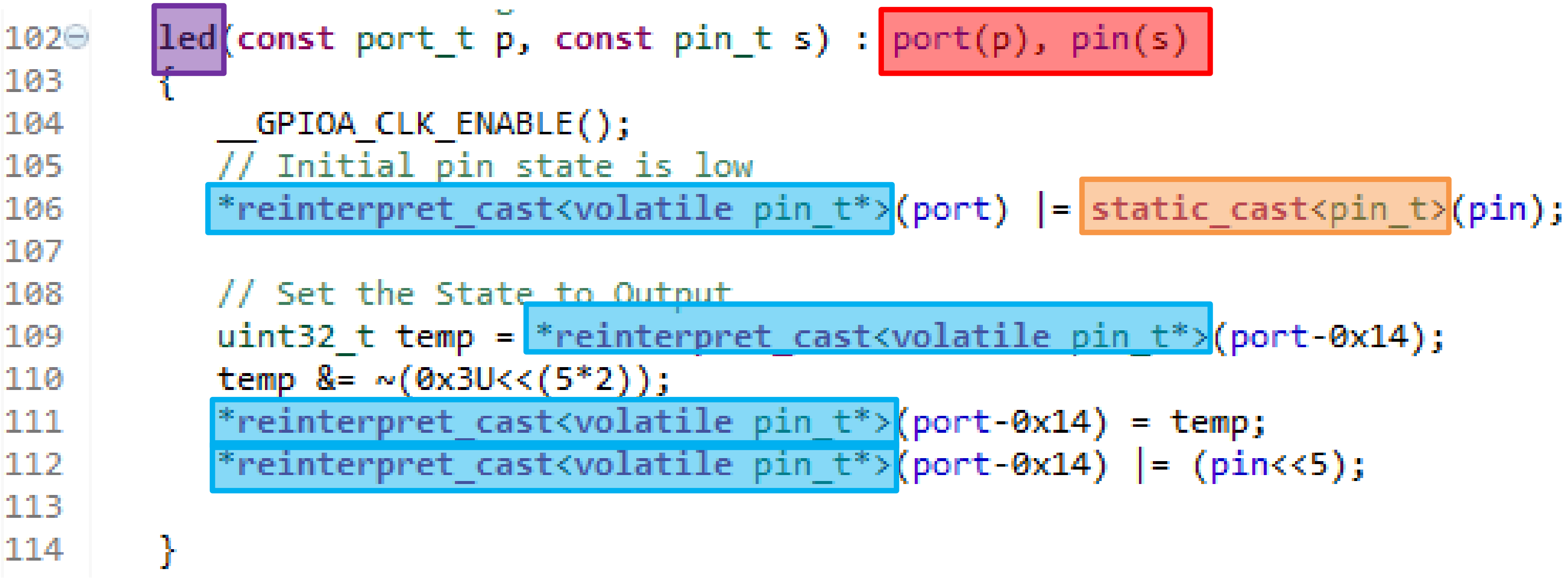
Methods

Private Data

Public Data and Methods

Private Data and Methods

The LED Class – Constructor

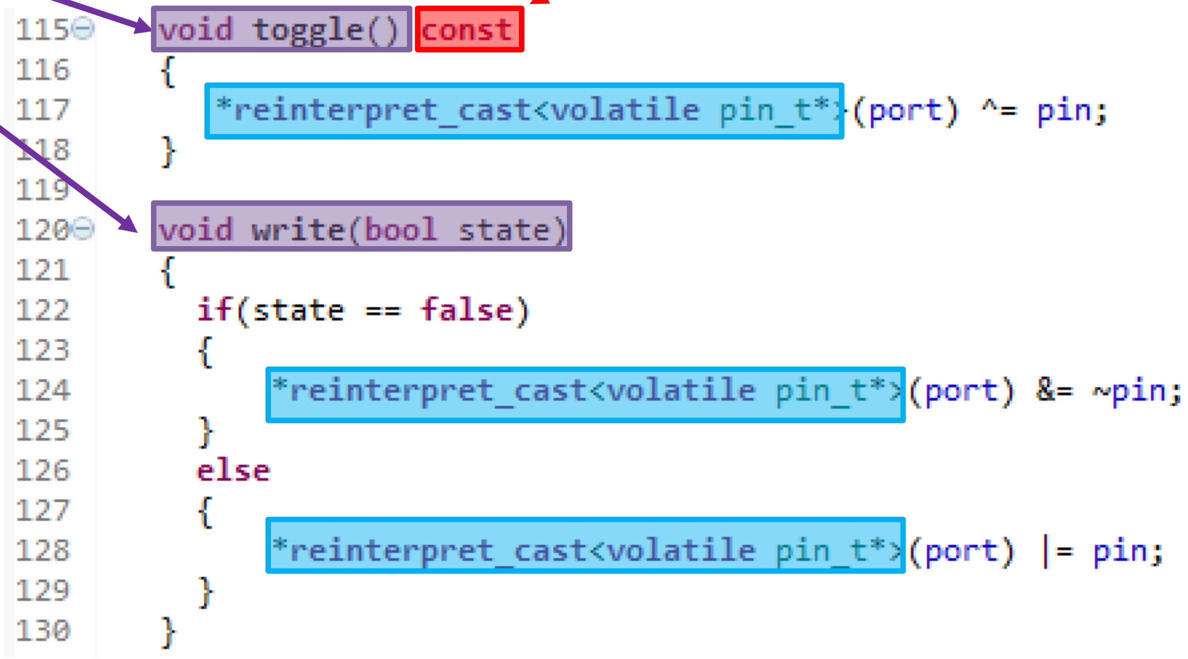


```
    led(const port_t p, const pin_t s) : port(p), pin(s)
    {
        __GPIOA_CLK_ENABLE();
        // Initial pin state is low
        *reinterpret_cast<volatile pin_t*>(port) |= static_cast<pin_t>(pin);

        // Set the State to Output
        uint32_t temp = *reinterpret_cast<volatile pin_t*>(port-0x14);
        temp &= ~(0x3U<<(5*2));
        *reinterpret_cast<volatile pin_t*>(port-0x14) = temp;
        *reinterpret_cast<volatile pin_t*>(port-0x14) |= (pin<<5);

    }
```

The LED Class - Methods

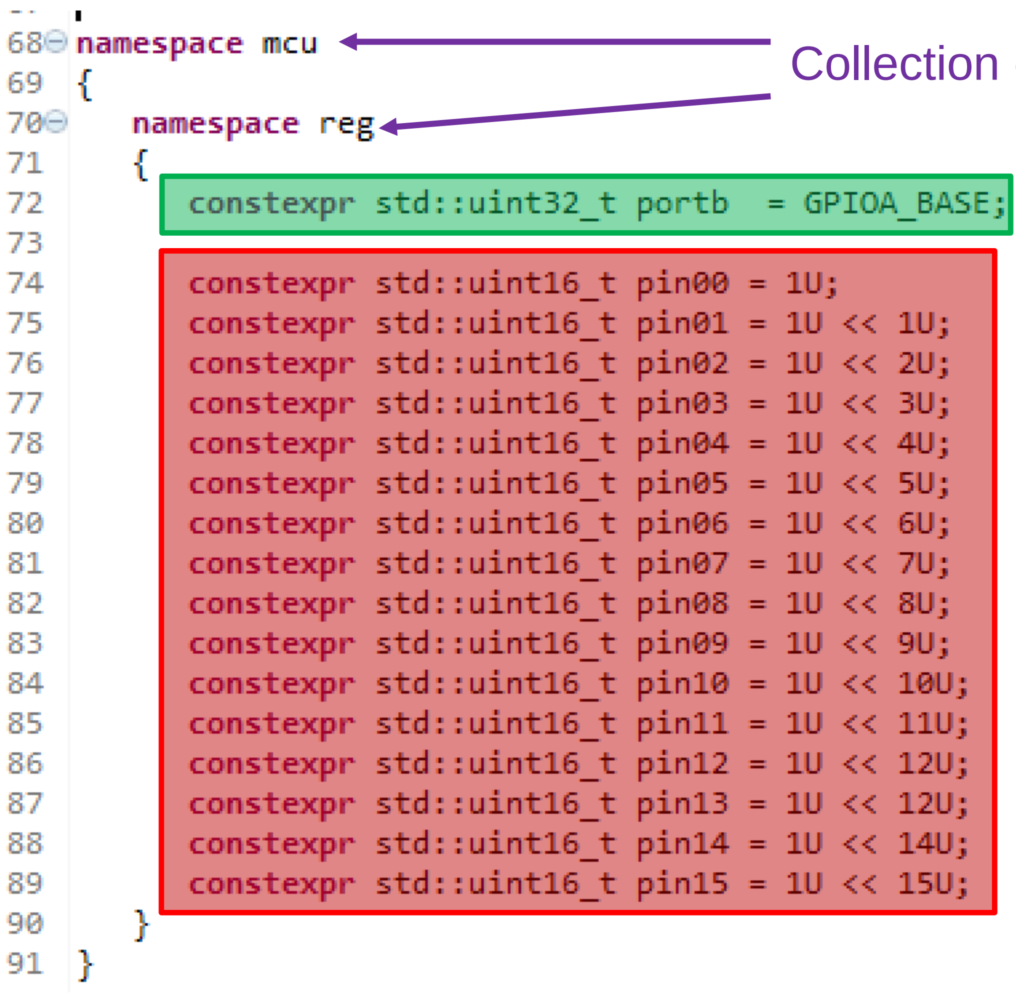
Methods

Cannot modify class member variables!



```
void toggle() const
{
   *reinterpret_cast<volatile pin_t*>(port) ^= pin;
}

void write(bool state)
{
  if(state == false)
  {
      *reinterpret_cast<volatile pin_t*>(port) &= ~pin;
  }
  else
  {
      *reinterpret_cast<volatile pin_t*>(port) |= pin;
  }
}
```

The Blinky LED Program - Setup



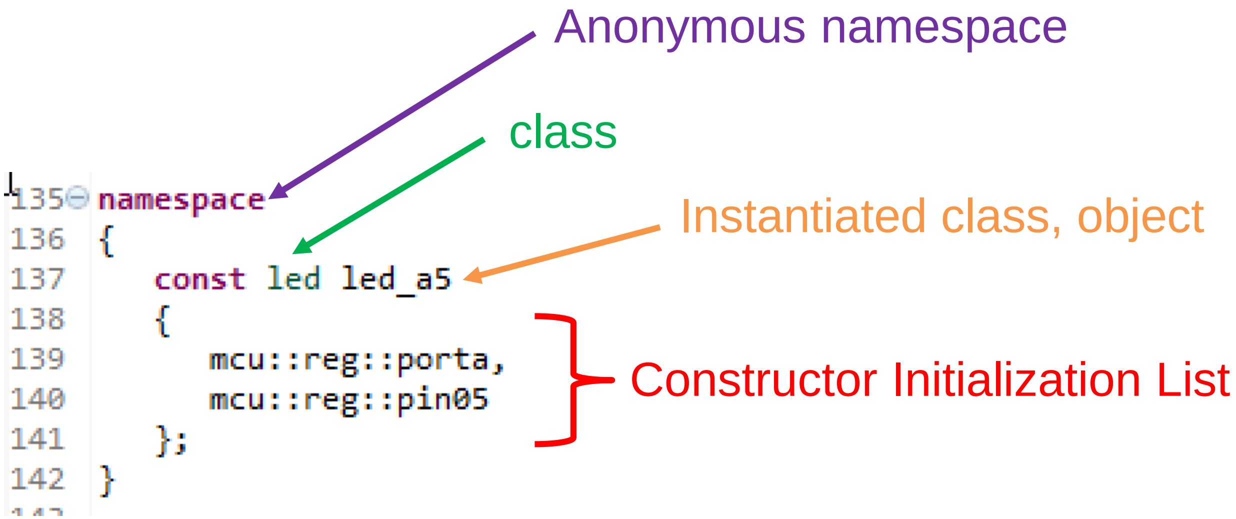
```
namespace mcu
{
    namespace reg
    {
        constexpr std::uint32_t portb  = GPIOA_BASE;

        constexpr std::uint16_t pin00 = 1U;
        constexpr std::uint16_t pin01 = 1U << 1U;
        constexpr std::uint16_t pin02 = 1U << 2U;
        constexpr std::uint16_t pin03 = 1U << 3U;
        constexpr std::uint16_t pin04 = 1U << 4U;
        constexpr std::uint16_t pin05 = 1U << 5U;
        constexpr std::uint16_t pin06 = 1U << 6U;
        constexpr std::uint16_t pin07 = 1U << 7U;
        constexpr std::uint16_t pin08 = 1U << 8U;
        constexpr std::uint16_t pin09 = 1U << 9U;
        constexpr std::uint16_t pin10 = 1U << 10U;
        constexpr std::uint16_t pin11 = 1U << 11U;
        constexpr std::uint16_t pin12 = 1U << 12U;
        constexpr std::uint16_t pin13 = 1U << 12U;
        constexpr std::uint16_t pin14 = 1U << 14U;
        constexpr std::uint16_t pin15 = 1U << 15U;
    }
}
```

Collection of related symbols

LED Port

Bit Masks

The Blinky LED Program Setup



```
namespace
{
    const led led_a5
    {
        mcu::reg::porta,
        mcu::reg::pin05
    };
}
```

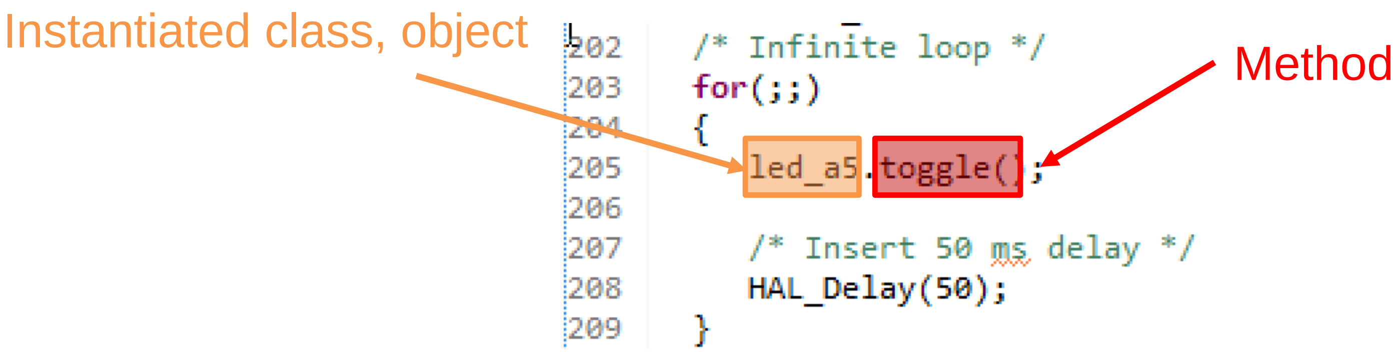
Anonymous namespace

class

Instantiated class, object

Constructor Initialization List

The Blinky LED Program

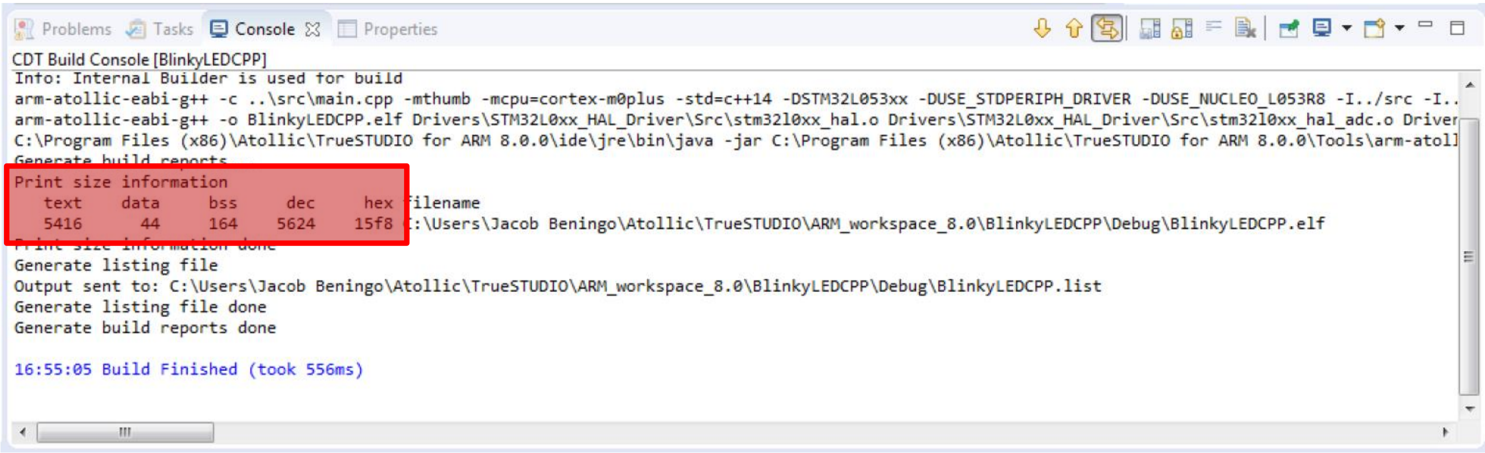


```
/* Infinite loop */
for(;;)
{
    led_a5.toggle();

    /* Insert 50 ms delay */
    HAL_Delay(50);
}
```

Instantiated class, object

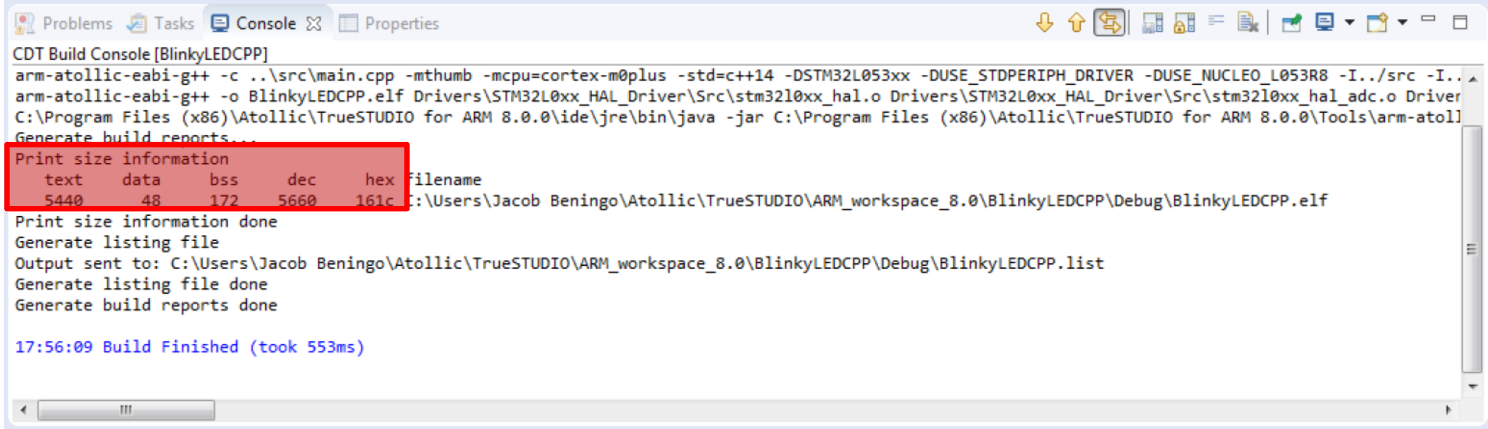
Method

C versus C++



CDT Build Console [BlinkyLEDCPP]

```
Info: Internal Builder is used for build
arm-atollic-eabi-g++ -c ..\src\main.cpp -mthumb -mcpu=cortex-m0plus -std=c++14 -DSTM32L053xx -DUSE_STDPERIPH_DRIVER -DUSE_NUCLEO_L053R8 -I../src -I..
arm-atollic-eabi-g++ -o BlinkyLEDCPP.elf Drivers\STM32L0xx_HAL_Driver\Src\stm32l0xx_hal.o Drivers\STM32L0xx_HAL_Driver\Src\stm32l0xx_hal_adc.o Driver
C:\Program Files (x86)\Atollic\TrueSTUDIO for ARM 8.0.0\ide\jre\bin\java -jar C:\Program Files (x86)\Atollic\TrueSTUDIO for ARM 8.0.0\Tools\arm-atoll
Generate build reports
Print size information
   text    data     bss     dec     hex filename
   5416      44     164    5624    15f8 C:\Users\Jacob Beningo\Atollic\TrueSTUDIO\ARM_workspace_8.0\BlinkyLEDCPP\Debug\BlinkyLEDCPP.elf
Print size information done
Generate listing file
Output sent to: C:\Users\Jacob Beningo\Atollic\TrueSTUDIO\ARM_workspace_8.0\BlinkyLEDCPP\Debug\BlinkyLEDCPP.list
Generate listing file done
Generate build reports done

16:55:05 Build Finished (took 556ms)
```



CDT Build Console [BlinkyLEDCPP]

```
arm-atollic-eabi-g++ -c ..\src\main.cpp -mthumb -mcpu=cortex-m0plus -std=c++14 -DSTM32L053xx -DUSE_STDPERIPH_DRIVER -DUSE_NUCLEO_L053R8 -I../src -I..
arm-atollic-eabi-g++ -o BlinkyLEDCPP.elf Drivers\STM32L0xx_HAL_Driver\Src\stm32l0xx_hal.o Drivers\STM32L0xx_HAL_Driver\Src\stm32l0xx_hal_adc.o Driver
C:\Program Files (x86)\Atollic\TrueSTUDIO for ARM 8.0.0\ide\jre\bin\java -jar C:\Program Files (x86)\Atollic\TrueSTUDIO for ARM 8.0.0\Tools\arm-atoll
Generate build reports...
Print size information
   text    data     bss     dec     hex filename
   5440      48     172    5660    161c C:\Users\Jacob Beningo\Atollic\TrueSTUDIO\ARM_workspace_8.0\BlinkyLEDCPP\Debug\BlinkyLEDCPP.elf
Print size information done
Generate listing file
Output sent to: C:\Users\Jacob Beningo\Atollic\TrueSTUDIO\ARM_workspace_8.0\BlinkyLEDCPP\Debug\BlinkyLEDCPP.list
Generate listing file done
Generate build reports done

17:56:09 Build Finished (took 553ms)
```

Presented by:

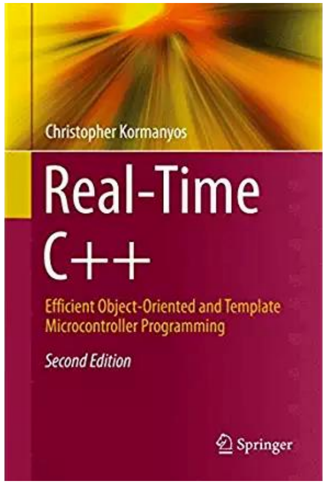
Transitioning to C++

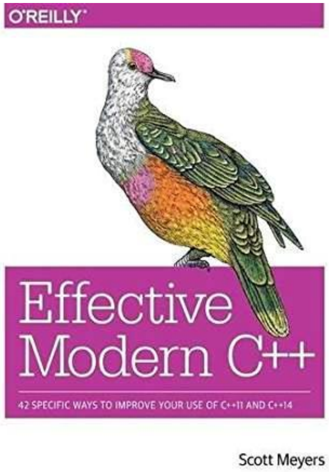
- Start simple
- Wrap existing code:

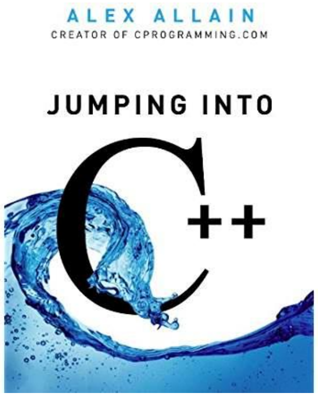
```
extern "C"
{
#include "main.h"
#include ”hardware.h"
}
```

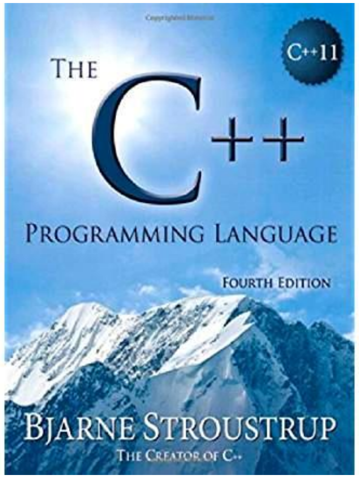
- Convert local static variables into anonymous namespaces
- Convert pointers to references
- Think objects and operations
- Avoid using the preprocessor!

Useful Resources


Christopher Kormanyos
Real-Time C++
Efficient Object-Oriented and Template Microcontroller Programming
Second Edition
Springer







Additional Resources

- Download Course Material for
  - C/C++ Doxygen Templates
  - Example source code
  - Blog
  - YouTube Videos
- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Transitioning from C to C++

The Lecturer – Jacob Beningo

Jacob Beningo

Principal Consultant

BENINGO EMBEDDED GROUP

Social Media / Contact

E : jacob@beningo.com

T : 810-844-1522

: Jacob_Beningo

f : Beningo Engineering

in : JacobBeningo

EDN : Embedded Basics

ARM Connected Community

Consulting

- Advising
- Coaching
- Content
- Consulting
- Training

www.beningo.com