Dependencies Between Layers for a Basic Console Output Component in $\mathsf{C}/\mathsf{C}{+}{+}$

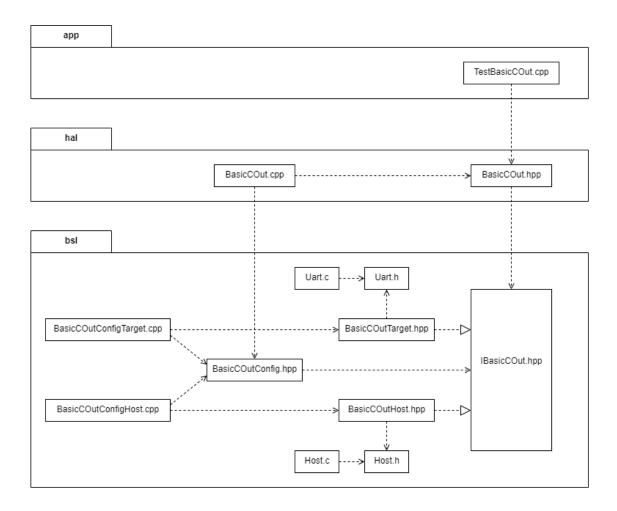
Dr. Michel de Champlain SOEN 422 — Embedded Systems and Software Fall 2021

November 17, 2021

Contents

1	Dep	endencies Between Layers in C++	2
	1.1	hal_TestBasicCOut.cpp	2
	1.2	hal_BasicCOut.hpp	3
	1.3	hal_BasicCOut.cpp	3
	1.4	bsl_IBasicCOut.hpp	3
	1.5	bsl_BasicCOutConfig.hpp	4
	1.6	bsl_BasicCOutHost.hpp	4
	1.7	bsl_BasicCOutTarget.hpp	5
	1.8	bsl_BasicCOutConfigHost.cpp	5
	1.9	bsl_BasicCOutConfigTarget.cpp	5
	1.10	bsl_Host.h	6
	1.11	bsl_Host.c	6
	1.12	bsl_Uart.h	6
	1.13	bsl_Uart.c	6
_	_		_
2	•	endencies Between Layers in C	7
	2.1	hal_TestBasicCOut.c	7
	2.2	hal_BasicCOut.h	8
	2.3	hal_BasicCOut.c	8
	2.4	bsl_IBasicCOut.h	8
	2.5	bsl_BasicCOutDesc.h	8
	2.6	bsl_BasicCOutConfig.h	9
	2.7	bsl_BasicCOutConfigHost.c	9
	2.8		10
	2.9		10
			10
			10
	2.12	bsl_Uart.c	10

1 Dependencies Between Layers in C++



Test in the Application layer in C++

1.1 hal_TestBasicCOut.cpp

```
// hal_TestBasicCOut.cpp - Generic Basic Console Output in C++.
    #include "hal_BasicCOut.hpp"
4
   int main (void) {
6
        hal::BasicCOut::Init(); // Hiding the factory and all dependencies with the host and targets.
7
8
        hal::BasicCOut::PutS("Test HAL Basic Console Output in C++\n");
        hal::BasicCOut::PutS("[t]\n"); // Expected output.
9
10
        hal::BasicCOut::PutC('[');
                                        // Current output.
11
        hal::BasicCOut::PutC('t');
12
13
        hal::BasicCOut::PutC(']');
14
        hal::BasicCOut::PutN();
15 }
```

HAL in C++

1.2 hal_BasicCOut.hpp

```
/* hal_BasicCOut.hpp - HAL Basic Console Output C++ Interface
3
   // Copyright (C) 1999-2021 by Michel de Champlain
   //
5
   */
6
7
   #ifndef __hal_BasicCOut_hpp
   #define __hal_BasicCOut_hpp
10 #include "bsl_IBasicCOut.hpp" // Need the interface declaration to declare a qualified opaque type.
11
12 class ::bsl::IBasicCOut;
                                 // Forward reference to the opaque type.
13
14 namespace hal {
       class BasicCOut {
15
16
       public:
17
           static void Init();
18
           static void PutC(char c);
           static void PutS(const char* s);
20
           static void PutN(void);
21
       private:
           static bsl::IBasicCOut* cout; // Opaque type for internal (BSL) console output.
22
23
           BasicCOut() = delete;
                                         // To never generate default constructor (C++11).
24
           ~BasicCOut() = delete;
                                         // To never generate default destructor (C++11).
25
       };
26 }
27
28 #endif
1.3 hal_BasicCOut.cpp
   /* hal_BasicCOut.cpp - HAL Basic Console Output C++ Interface
   //
   // Copyright (C) 1999-2021 by Michel de Champlain
3
   //
5
   */
   #include "hal_BasicCOut.hpp"
7
  #include "bsl_BasicCOutConfig.hpp"
10 void hal::BasicCOut::Init()
                                            { if (cout == nullptr) cout = bsl::BasicCOutConfig::Get(); }
11
12 void hal::BasicCOut::PutC(char c)
                                           { cout->PutC(c); }
13 void hal::BasicCOut::PutS(const char* s) { cout->PutS(s); }
14 void hal::BasicCOut::PutN(void)
                                            { cout->PutN(); }
15
16 bsl::IBasicCOut* hal::BasicCOut::cout = nullptr;
BSL in C++
1.4 bsl_IBasicCOut.hpp
   /* bsl_IBasicCOut.hpp - BSL Basic Console Output C++ Interface
   //
3
   // Copyright (C) 1999-2021 by Michel de Champlain
4
   //
   */
   #ifndef __bsl_IBasicCOut_hpp
```

```
8
   #define __bsl_IBasicCOut_hpp
10 namespace bsl {
11
       class IBasicCOut {
       public:
12
13 //
           void virtual Init(); // No Ctor() or Init() in an (emulated) interface in C++.
14
           void virtual PutC(char c) = 0;
           void virtual PutS(const char* s) = 0;
15
16
           void virtual PutN(void) = 0;
17
18 }
19 #endif
1.5 bsl_BasicCOutConfig.hpp
    /* bsl_BasicCOutConfig.hpp - BSL Basic Console Output Config (Static Factory Method) C++ Interface
   //
                                This is a unique and uniform interface for all BSL BasicCOut configurations.
3
   // Copyright (C) 1999-2021 by Michel de Champlain
   //
6
   */
8
   #ifndef __bsl_BasicCOutConfig_hpp
  #define __bsl_BasicCOutConfig_hpp
10
11 #include "bsl_IBasicCOut.hpp"
12
13
   namespace bsl {
14
       class BasicCOutConfig {
15
           static IBasicCOut* Get(); // Static Factory method
16
17
18 }
19 #endif
1.6 bsl_BasicCOutHost.hpp
    /* bsl_BasicCOutHost.hpp - BSL Basic Console Output Host C++ Interface (inlining the class)
1
2
   // Note: In this header file (.hpp), I implicitly declare inline functions in the class declaration by
   // simply defining (implementing) functions inside the class declaration. No inline keyword is necessary.
   // It is a way to give hints to the C++ compiler to consider optimizing these functions as macros.
   //
   // Copyright (C) 1999-2021 by Michel de Champlain
7
8
   //
9
10
   #ifndef __bsl_BasicCOutHost_hpp
11
12 #define __bsl_BasicCOutHost_hpp
13
14 extern "C" {
15 #include "bsl_Host.h"
16 }
17
18 #include "bsl_IBasicCOut.hpp"
19
20 namespace bsl {
21
       class BasicCOutHost : public IBasicCOut {
22
       public:
                        BasicCOutHost()
                                            { bsl_Host_Init(); }
24
           void virtual PutC(char c)
                                            { bsl_Host_PutC(c); }
25
           void virtual PutS(const char* s) { while (*s) bsl_Host_PutC(*s++); }
           void virtual PutN()
                                            { bsl_Host_PutC('\n'); }
```

```
27
       };
28 }
29 #endif
1.7 bsl_BasicCOutTarget.hpp
   /* bsl_BasicCOutTarget.hpp - BSL Basic Console Output Target C++ (inlining the class)
2
   // Note: In this header file (.hpp), I implicitly declare inline functions in the class declaration by
3
   // simply defining (implementing) functions inside the class declaration. No inline keyword is necessary.
   // It is a way to give hints to the C++ compiler to consider optimizing these functions as macros.
6
   // Copyright (C) 1999-2021 by Michel de Champlain
8
  //
9
   */
10
11 #ifndef __bsl_BasicCOutTarget_hpp
12 #define __bsl_BasicCOutTarget_hpp
13
14 extern "C" {
15 #include "bsl_Uart.h"
16 }
17
18 #include "bsl_IBasicCOut.hpp"
19
20 namespace bsl {
21
       class BasicCOutTarget : public IBasicCOut {
22
       public:
                        BasicCOutTarget()
                                           { bsl_Uart_Init(); }
24
            void virtual PutC(char c)
                                            { bsl_Uart_TxChar(c); }
25
           void virtual PutS(const char* s) { while (*s) bsl_Uart_TxChar(*s++); }
26
            void virtual PutN()
                                            { bsl_Uart_TxChar('\n'); }
27
       };
28 }
29 #endif
1.8 bsl_BasicCOutConfigHost.cpp
    /* bsl_BasicCOutConfigHost.cpp - BSL Basic Console Output Config (Factory Method) C++ Implementation
   //
2
                                    Note: Can be compiled separately, but link only one at a time.
3
   //
4
   // Copyright (C) 1999-2021 by Michel de Champlain
   //
8
   #include "bsl_BasicCOutConfig.hpp"
9
   #include "bsl_BasicCOutHost.hpp"
10
11 bsl::IBasicCOut* bsl::BasicCOutConfig::Get() { return new bsl::BasicCOutHost(); }
1.9 bsl_BasicCOutConfigTarget.cpp
    /* bsl_BasicCOutConfigTarget.cpp - BSL Basic Console Output Config for Target (Factory Method) C++ Implementation
   //
                                      Note: Can be compiled separately, but link only one at a time.
3
4
   // Copyright (C) 1999-2021 by Michel de Champlain
   //
6
  #include "bsl_BasicCOutConfig.hpp"
   #include "bsl_BasicCOutTarget.hpp"
11 bsl::IBasicCOut* bsl::BasicCOutConfig::Get() { return new bsl::BasicCOutTarget(); }
```

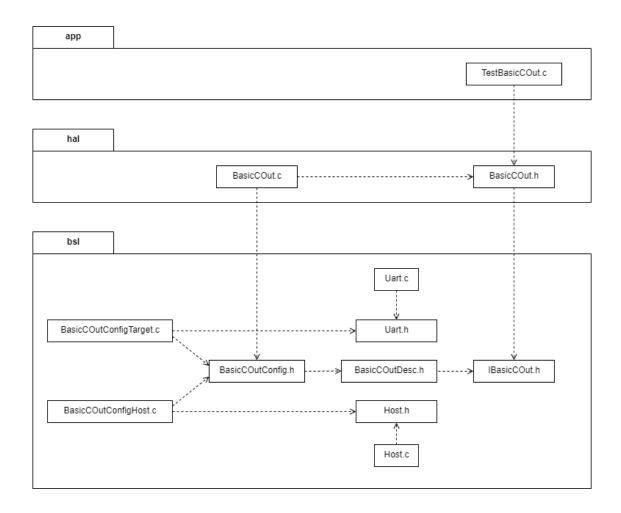
BSL in C

1.10 bsl_Host.h

```
/* bsl_Host.h - Host interface for Console Output
   // Copyright (C) 2020-2021 by Michel de Champlain
3
   //
5
6
7
   #ifndef __bsl_Host_h
8 #define __bsl_Host_h
10 void bsl_Host_Init(void);
11 void bsl_Host_PutC(char c);
12 char bsl_Host_GetC(void);
13
14 #endif
1.11 bsl_Host.c
    /*\ \mathtt{bsl\_Host.c}\ \mathtt{-}\ \mathtt{Host}\ \mathtt{implementation}\ \mathtt{for}\ \mathtt{Console}\ \mathtt{Output}
   // Copyright (C) 2020-2021 by Michel de Champlain
   //
5
   */
   #include <stdio.h>
9 void bsl_Host_Init(void) { }
10 void bsl_Host_PutC(char c) { putchar(c); }
11 char bsl_Host_GetC(void) { return getchar(); }
1.12 bsl_Uart.h
   /* bsl_Uart.h - Uart header for Arduino Nano
  //
3
   // Copyright (C) 2020-2021 by Michel de Champlain
   //
4
   */
6
   #ifndef __bsl_Uart_h
7
   #define __bsl_Uart_h
10 void bsl_Uart_Init(void);
11 void bsl_Uart_TxChar(char c);
12 char bsl_Uart_RxChar(void);
13
14 #endif
```

1.13 bsl_Uart.c

2 Dependencies Between Layers in C



Test in the Application layer in C

2.1 hal_TestBasicCOut.c

```
// hal_TestBasicCOut.c - Generic Basic Console Output in C
3
    #include "hal_BasicCOut.h"
4
    int main (void) {
                                  \ensuremath{//} Hiding the factory and all dependencies with the host and targets.
6
        hal_COut_Init();
8
        hal_COut_PutS("Test HAL Basic Console Output in C\n");
        hal_COut_PutS("[t]\n"); // Expected output.
9
10
        hal_COut_PutC('[');
                                  // Current output.
11
12
        hal_COut_PutC('t');
        hal_COut_PutC(']');
13
14
        hal_COut_PutN();
15 }
```

HAL in C

2.2 hal_BasicCOut.h

```
/* hal_BasicCOut.h - HAL Basic Console Output Interface
3
  // Copyright (C) 1999-2021 by Michel de Champlain
  //
5
  */
6
7
  #ifndef __hal_BasicCOut_h
8 #define __hal_BasicCOut_h
10 void hal_COut_Init(void);
11 void hal_COut_PutC(char c);
12 void hal_COut_PutS(const char* s);
13 void hal_COut_PutN(void);
15 #endif
2.3 hal_BasicCOut.c
   /* hal_BasicCOut.c - HAL Basic Console Output Implementation
  // Copyright (C) 1999-2021 by Michel de Champlain
3
  //
4
5
  #include "hal_BasicCOut.h"
7
                                     // Basic Output Descriptor.
8 //#include "bsl_BasicCOutDesc.h"
  #include "bsl_BasicCOutConfig.h" // bsl_ICOut opaque type and the factory method.
10
11 static bsl_IBasicCOut cout;
13 void hal_COut_Init(void)
                                    { cout = bsl_BasicCOut_Get(); }
14 void hal_COut_PutC(char c)
                                   { cout->pc(c); }
void hal_COut_PutS(const char* s) { cout->ps(s); }
16 void hal_COut_PutN(void)
                             { cout->pn(); }
BSL in C
2.4 bsl_IBasicCOut.h
   /* bsl_IBasicCOut.h - BSL for Console Output Interface to hide the internal descriptor.
   //
3
   // Copyright (C) 1999-2020 by Michel de Champlain
   //
5
   */
  #ifndef __bsl_IBasicCOut_h
8 #define __bsl_IBasicCOut_h
           struct bsl_BasicCOutDesc;
10
11 typedef struct bsl_BasicCOutDesc* bsl_IBasicCOut;
12
13 #endif
2.5 bsl_BasicCOutDesc.h
```

```
1  /* bsl_BasicCOutDesc.h - Interface for Basic Output Descriptor
2  //
3  // Copyright (c) 1999-2021 by Michel de Champlain
```

```
4
   //
5
6
7
   #ifndef __bsl_BasicCOutDesc_h
   #define __bsl_BasicCOutDesc_h
8
10 #include "bsl_IBasicCOut.h"
11
12 // Private Basic Output Function Pointer Types:
13 typedef void (*bsl_PutC)(char);
14 typedef void (*bsl_PutS)(const char*);
15 typedef void (*bsl_PutN)(void);
17 // Private Interface Basic Output Descriptor
18 typedef struct bsl_BasicCOutDesc {
19
       bsl_PutC pc;
       bsl_PutS ps;
20
21
       bsl_PutN pn;
23
24 #endif
2.6 bsl_BasicCOutConfig.h
   /* bsl_BasicCOutConfig.h - BSL for Basic Console Output Config (Static Factory Method) C Interface
   //
2
                              This is a unique and uniform interface for all BSL BasicCOut configurations.
3
   //
   // Copyright (C) 1999-2021 by Michel de Champlain
   //
6
   */
7
8
   #ifndef __bsl_BasicCOutConfig_h
   #define __bsl_BasicCOutConfig_h
9
10
                                          // Basic Output Descriptor which in turn depends on {\tt IBasicCOut.h}
#include "bsl_BasicCOutDesc.h"
12
13 bsl_IBasicCOut bsl_BasicCOut_Get(void); // Static Factory Method (Function)
14
15 #endif
2.7 bsl_BasicCOutConfigHost.c
   /* bsl_BasicCOutConfigHost.c - Basic Console Output Configuration implementation for Host.
   //
3
   // Copyright (C) 1999-2021 by Michel de Champlain
   //
5
   */
6
   #include "bsl_BasicCOutConfig.h" // Depends of BasicCOutDesc.h
7
8
  #include "bsl_Host.h"
10 static void COut_PutC(char c)
                                       { bsl_Host_PutC(c); }
   static void COut_PutS(const char* s) { while (*s) bsl_Host_PutC(*s++); }
12 static void COut_PutN(void)
                                        { bsl_Host_PutC('\n'); }
13
14
   static bsl_BasicCOutDesc cout = {
15
       COut_PutC,
16
       COut_PutS,
17
       COut_PutN
18 };
19
20 bsl_IBasicCOut bsl_BasicCOut_Get(void) { return &cout; }
```

2.8 bsl_BasicCOutConfigTarget.c

2.9 bsl_Host.h

```
/* bsl_Host.h - Host interface for Console Output
   // Copyright (C) 2020-2021 by Michel de Champlain
  //
5
6
7
  #ifndef __bsl_Host_h
  #define __bsl_Host_h
10 void bsl_Host_Init(void);
11 void bsl_Host_PutC(char c);
12 char bsl_Host_GetC(void);
13
14 #endif
2.10 bsl_Host.c
   /*\ \mbox{bsl\_Host.c} - Host implementation for Console Output
   // Copyright (C) 2020-2021 by Michel de Champlain
5
   */
   #include <stdio.h>
9 void bsl_Host_Init(void) { }
10 void bsl_Host_PutC(char c) { putchar(c); }
11 char bsl_Host_GetC(void) { return getchar(); }
2.11 bsl_Uart.h
   /* bsl_Uart.h - Uart header for Arduino Nano
  //
   // Copyright (C) 2020-2021 by Michel de Champlain
   //
4
   */
7
   #ifndef __bsl_Uart_h
  #define __bsl_Uart_h
10 void bsl_Uart_Init(void);
11 void bsl_Uart_TxChar(char c);
12 char bsl_Uart_RxChar(void);
14 #endif
```

2.12 bsl_Uart.c