



DesignNews

Embedded Studio Primer

DAY 1: Teensy Embedded Studio Conversion

Sponsored by



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.



Fred Eady

Visit 'Lecturer Profile' in your console for more details.

Embedded Studio Primer

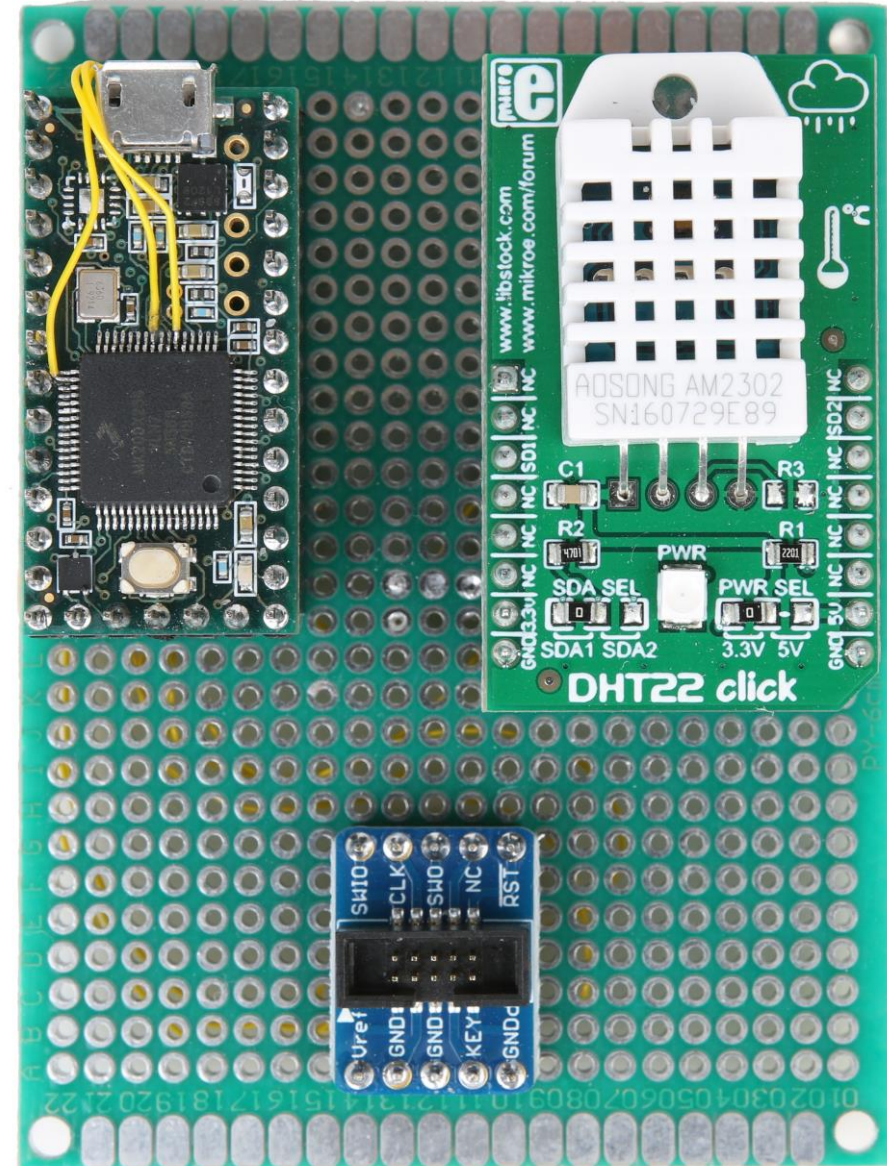
Teensy Embedded Studio Conversion

The screenshot displays the Embedded Studio IDE interface for a Teensy project. The main window is split into several panes:

- Disassembly:** Shows assembly code for the `main` function, including instructions like `PTC->PCOR = 0x00000020;` and `PORTD->PCR[1] = PORT_PCR_MUX(1) | PORT_PCR_DSE_MASK;`.
- Source Code:** Shows the C code for `init(void)` and `main(void)`. The `init` function configures various hardware modules like SCGC5, SCGC6, PORTB, UART0, and PORTC. The `main` function calls `startPulse()` and `get40bits()`.
- Registers:** A window titled "emStudio" showing a list of registers and their values. The `UART0_C4` register is highlighted with a value of `0x18`.
- Output:** Shows the progress of downloading the target script file and the Teensy project ELF file to the target device.

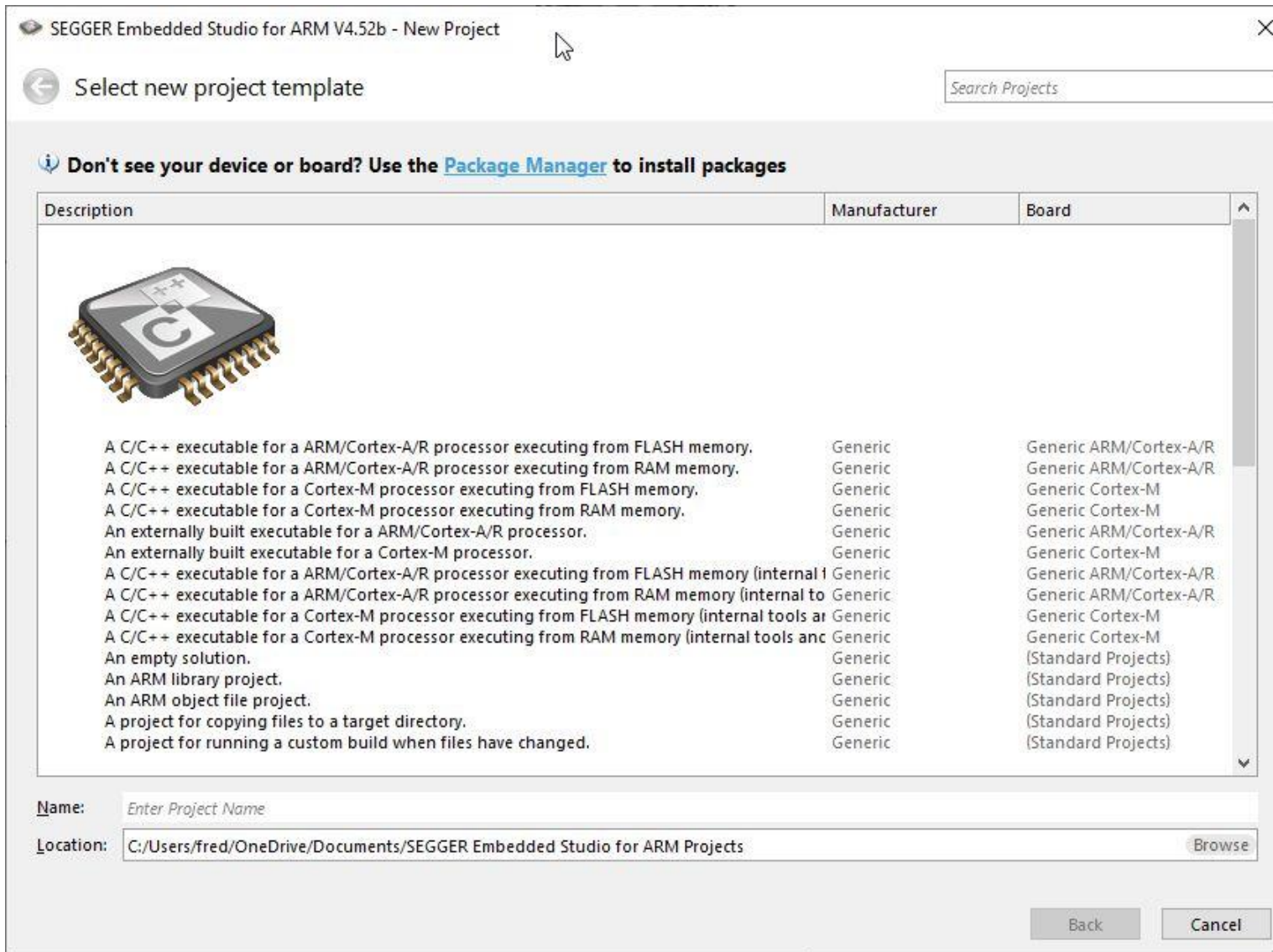
AGENDA

- **MK20D7 Project Creation**
- **Teensy 3.2 Hardware Modification**
- **Embedded Studio Fly-Over**
- **Coding a DHT22 Application**



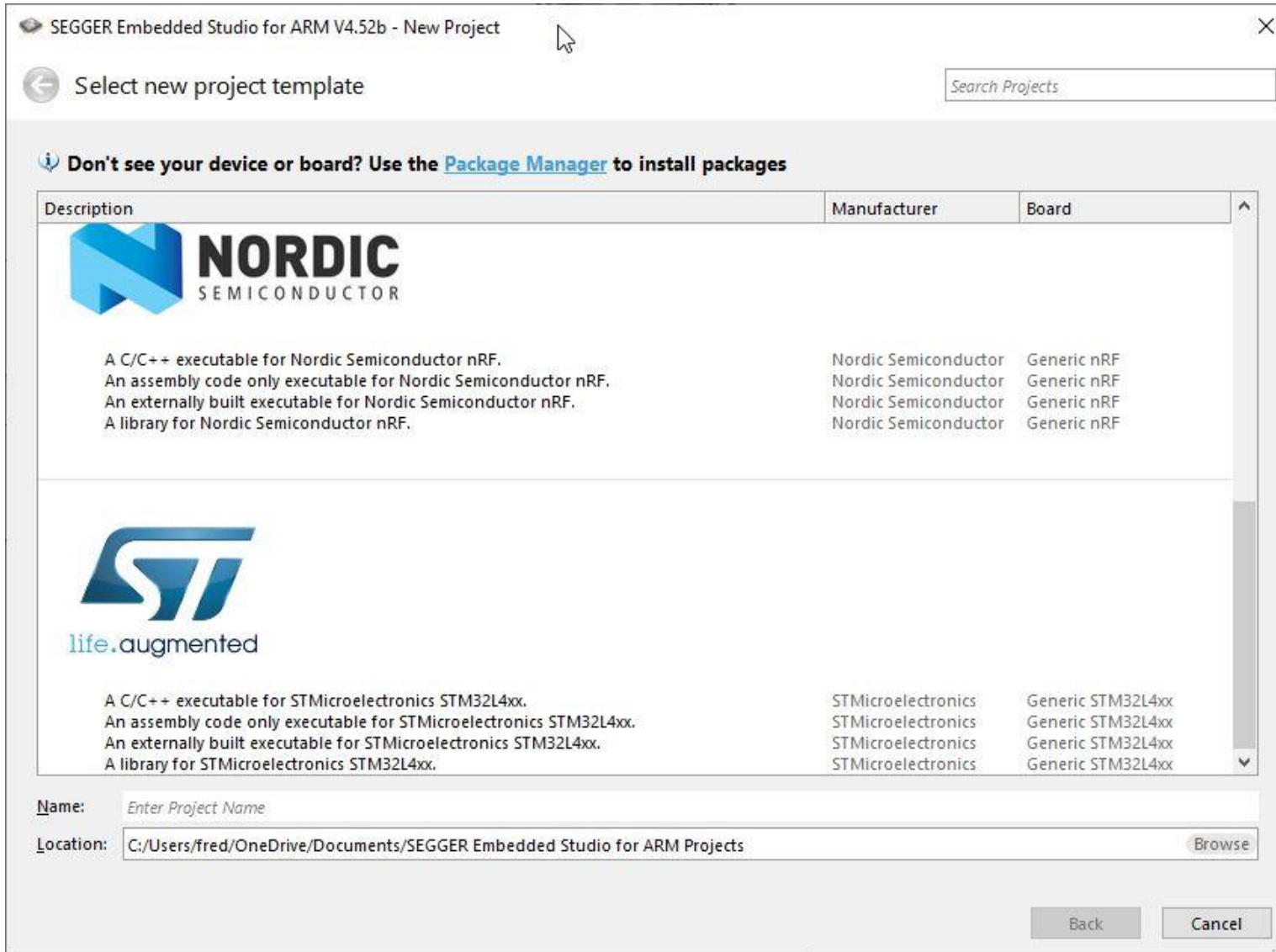
Embedded Studio Primer

MK20D7 Project Creation



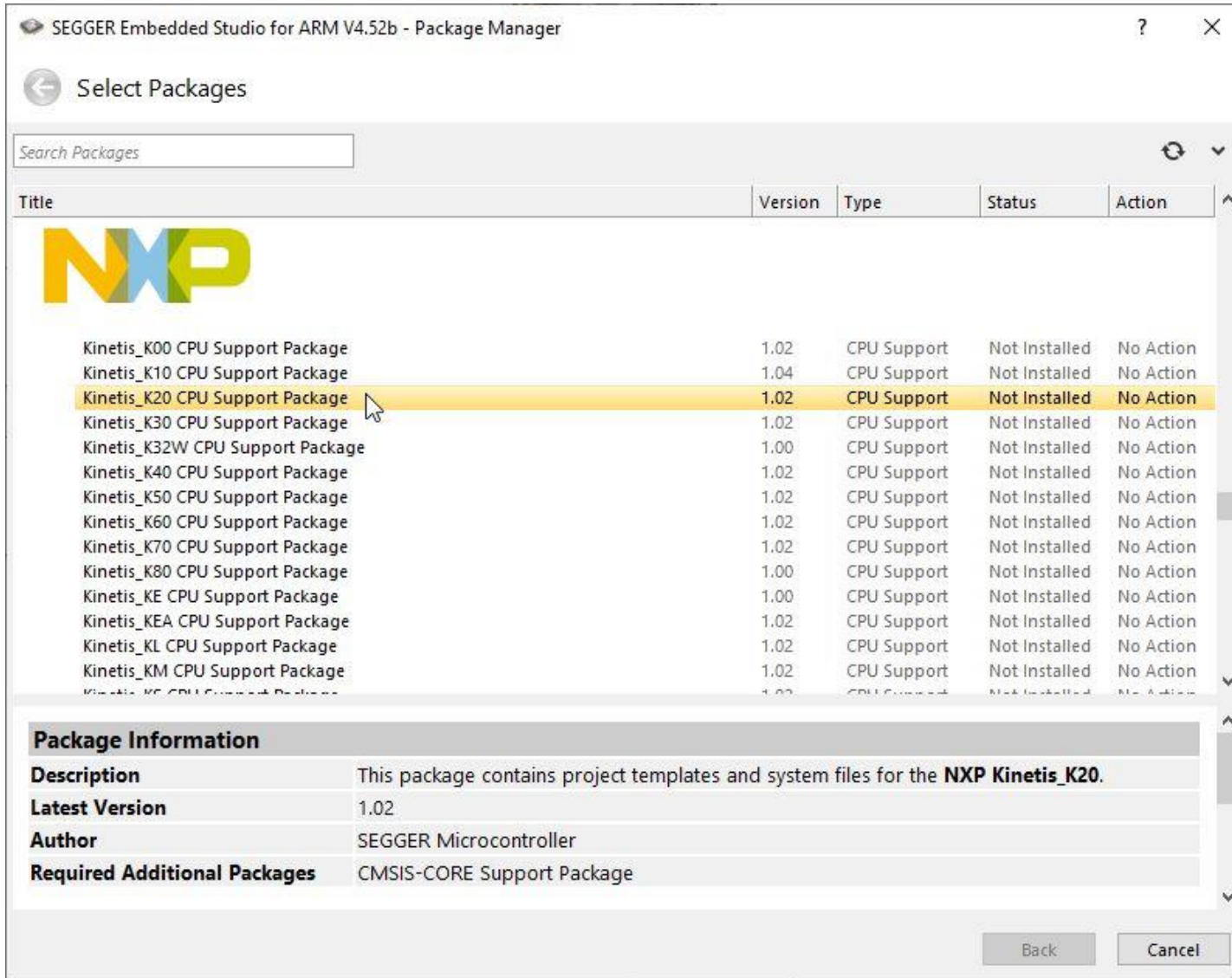
Embedded Studio Primer

MK20D7 Project Creation



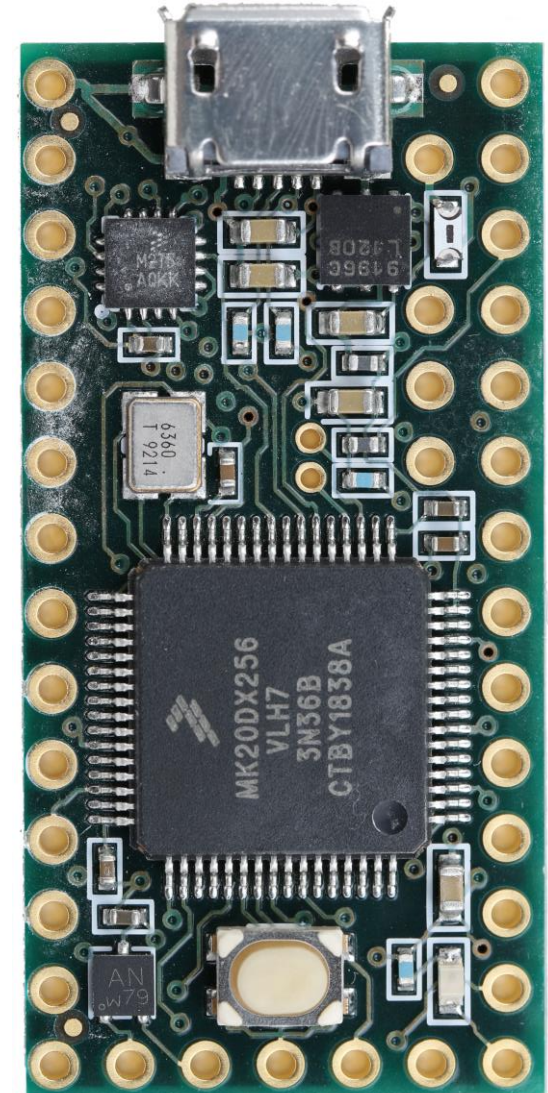
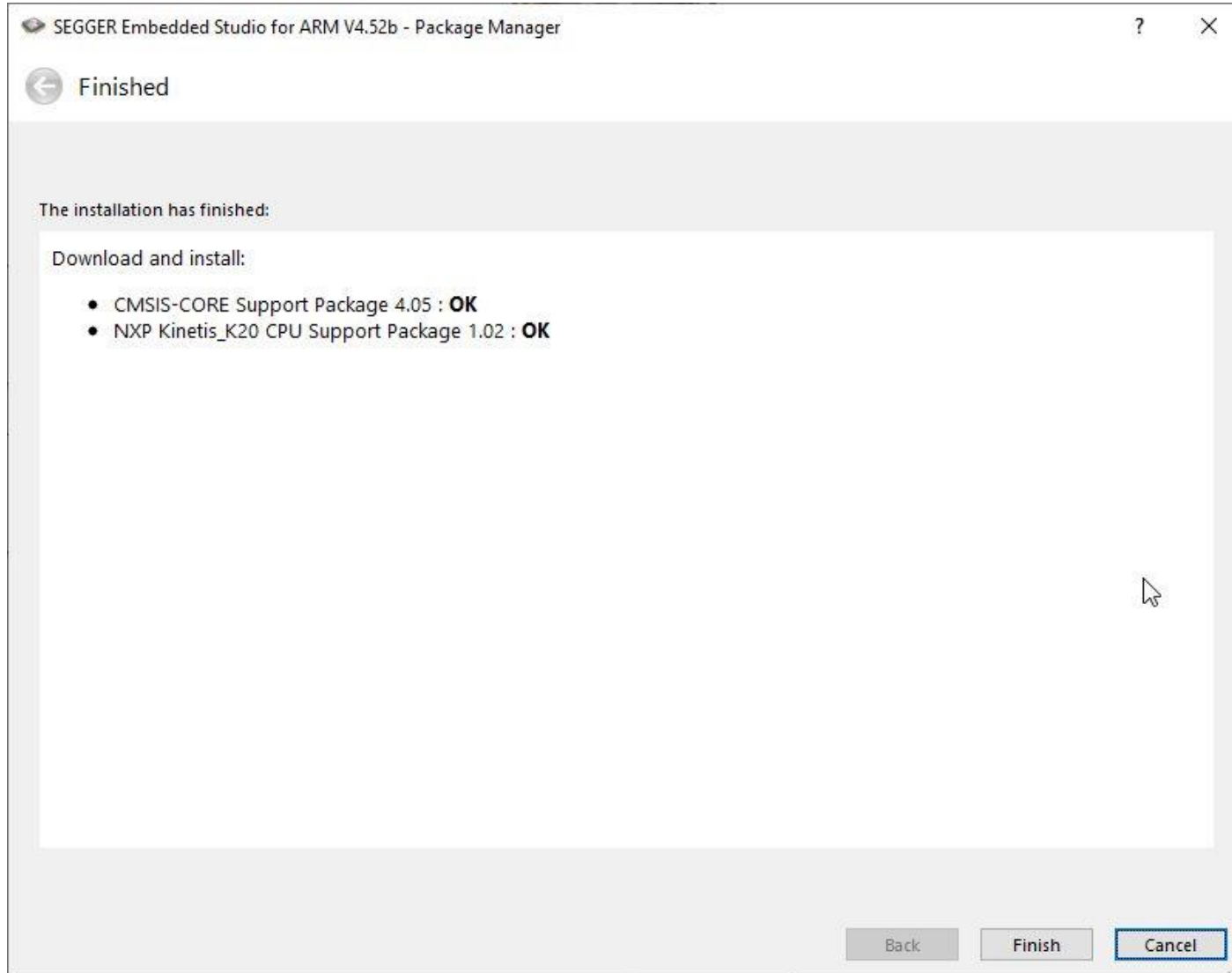
Embedded Studio Primer

MK20D7 Project Creation



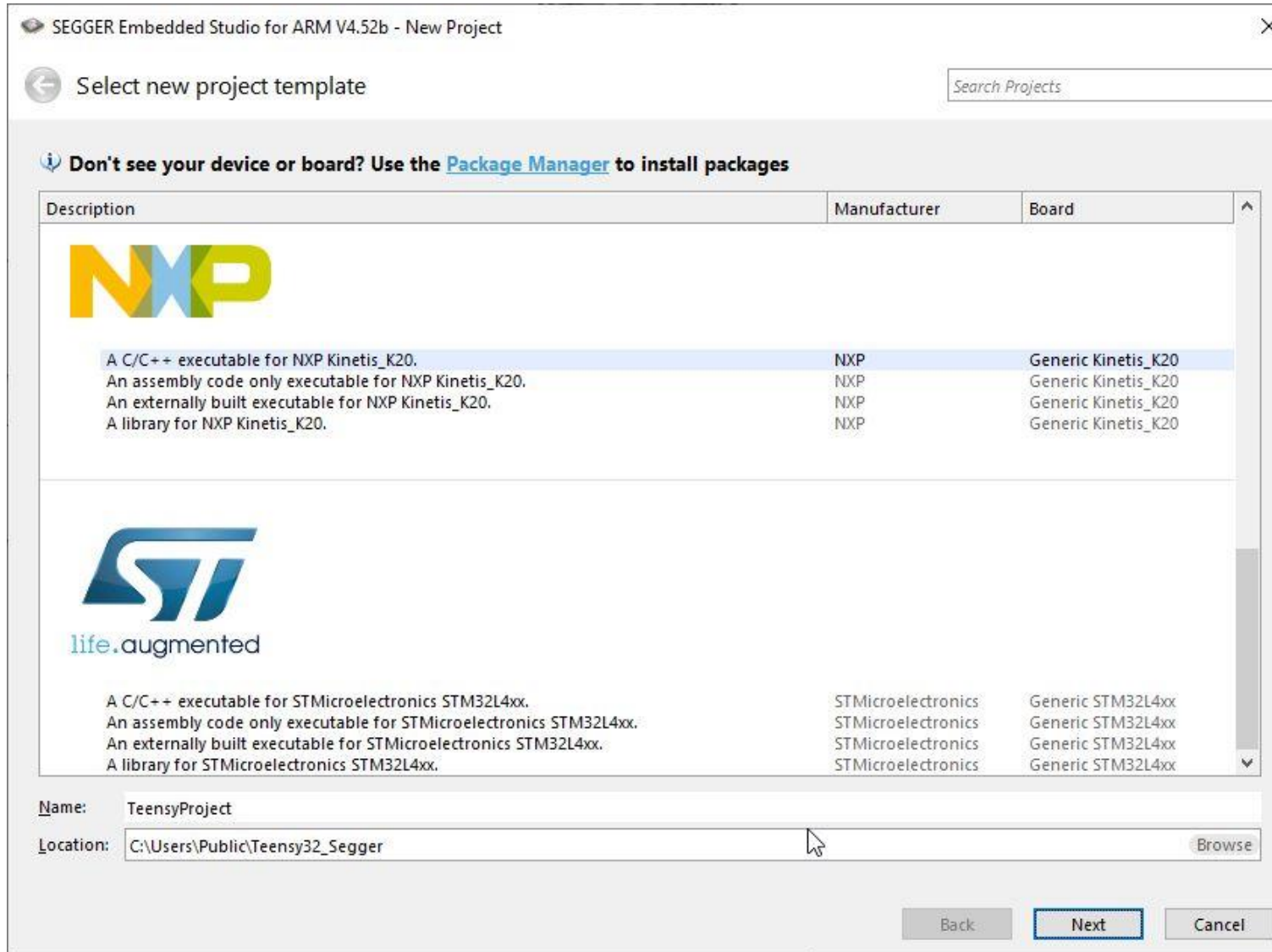
Embedded Studio Primer

MK20D7 Project Creation



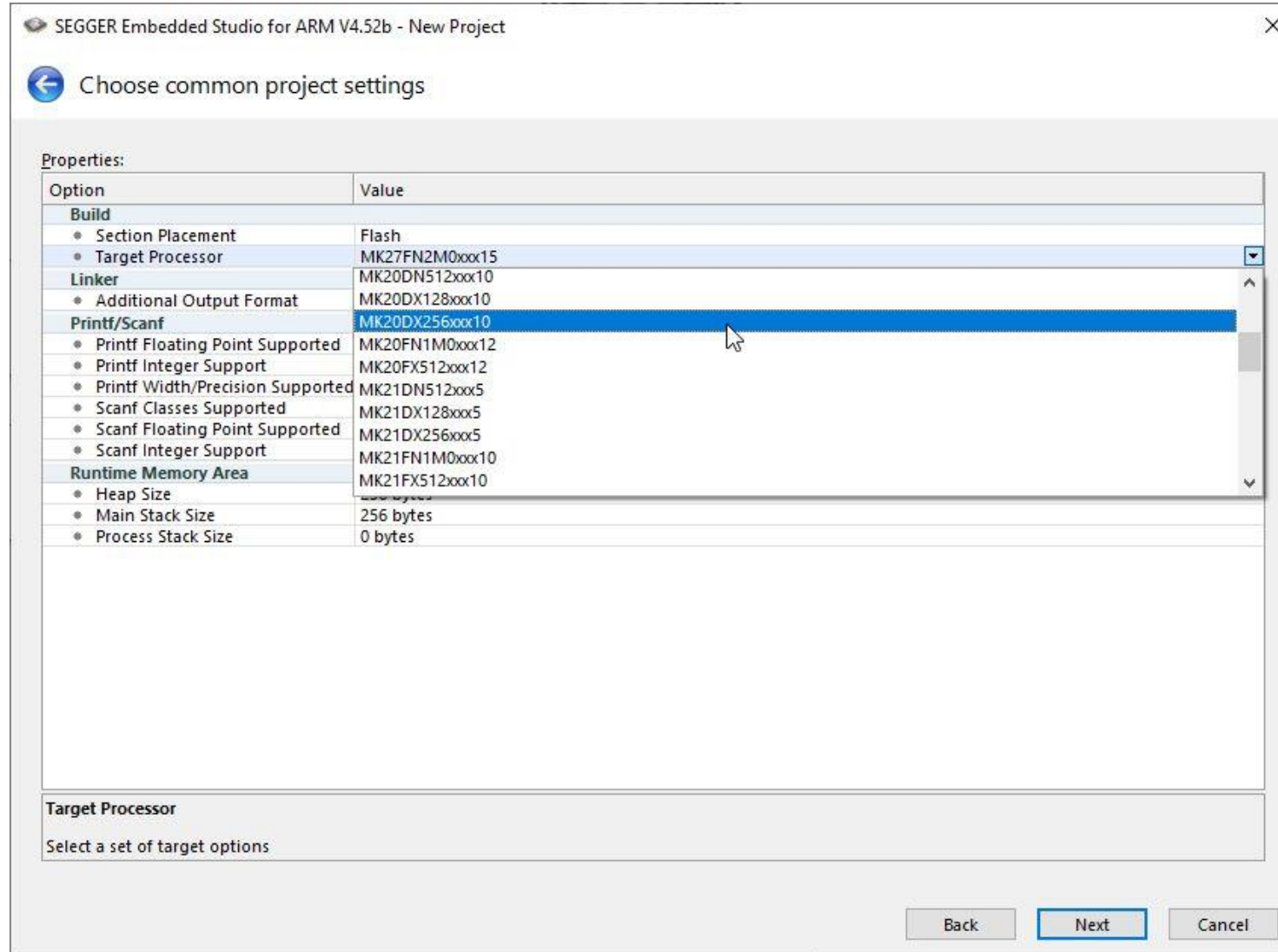
Embedded Studio Primer

MK20D7 Project Creation



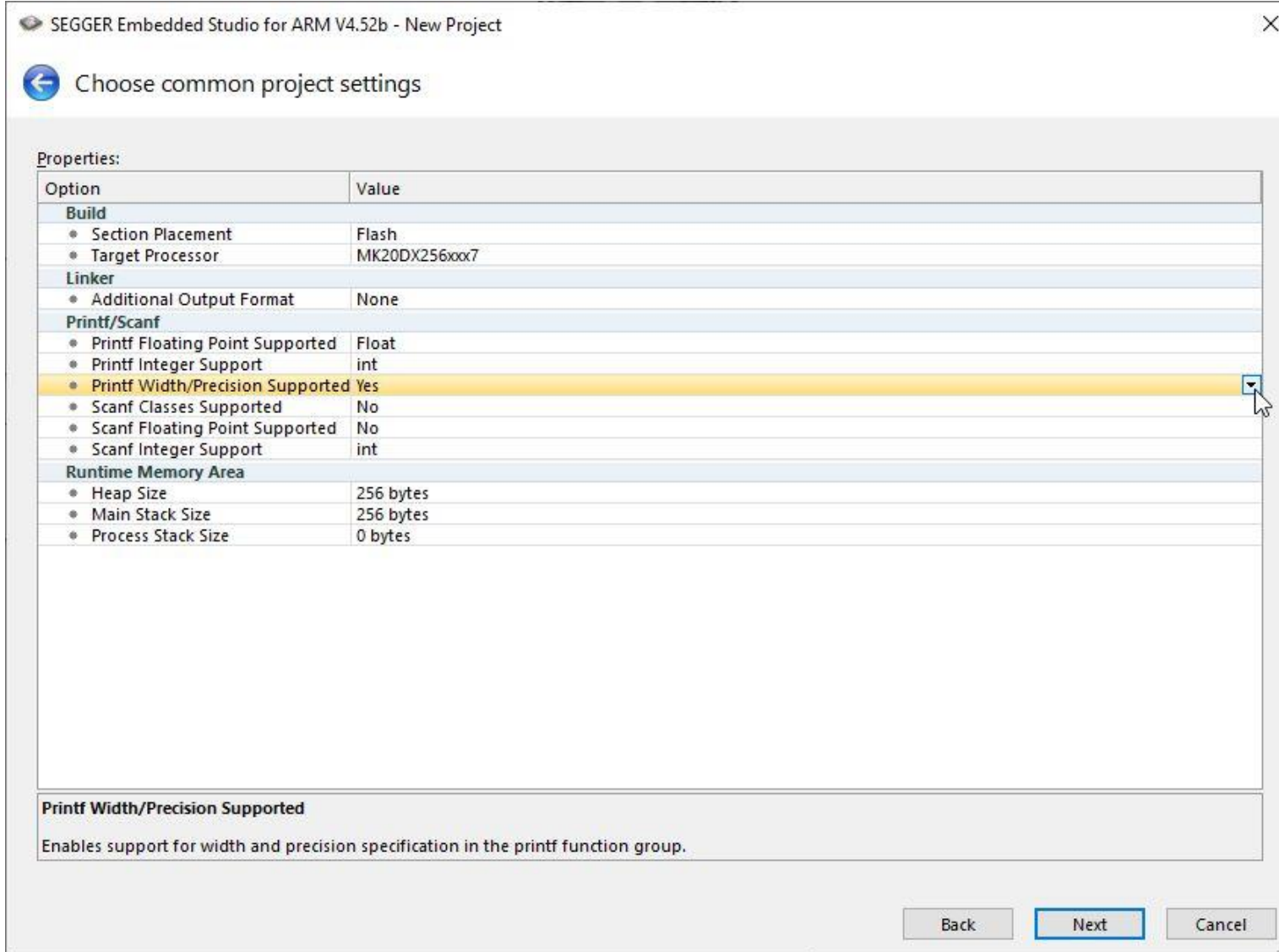
Embedded Studio Primer

MK20D7 Project Creation



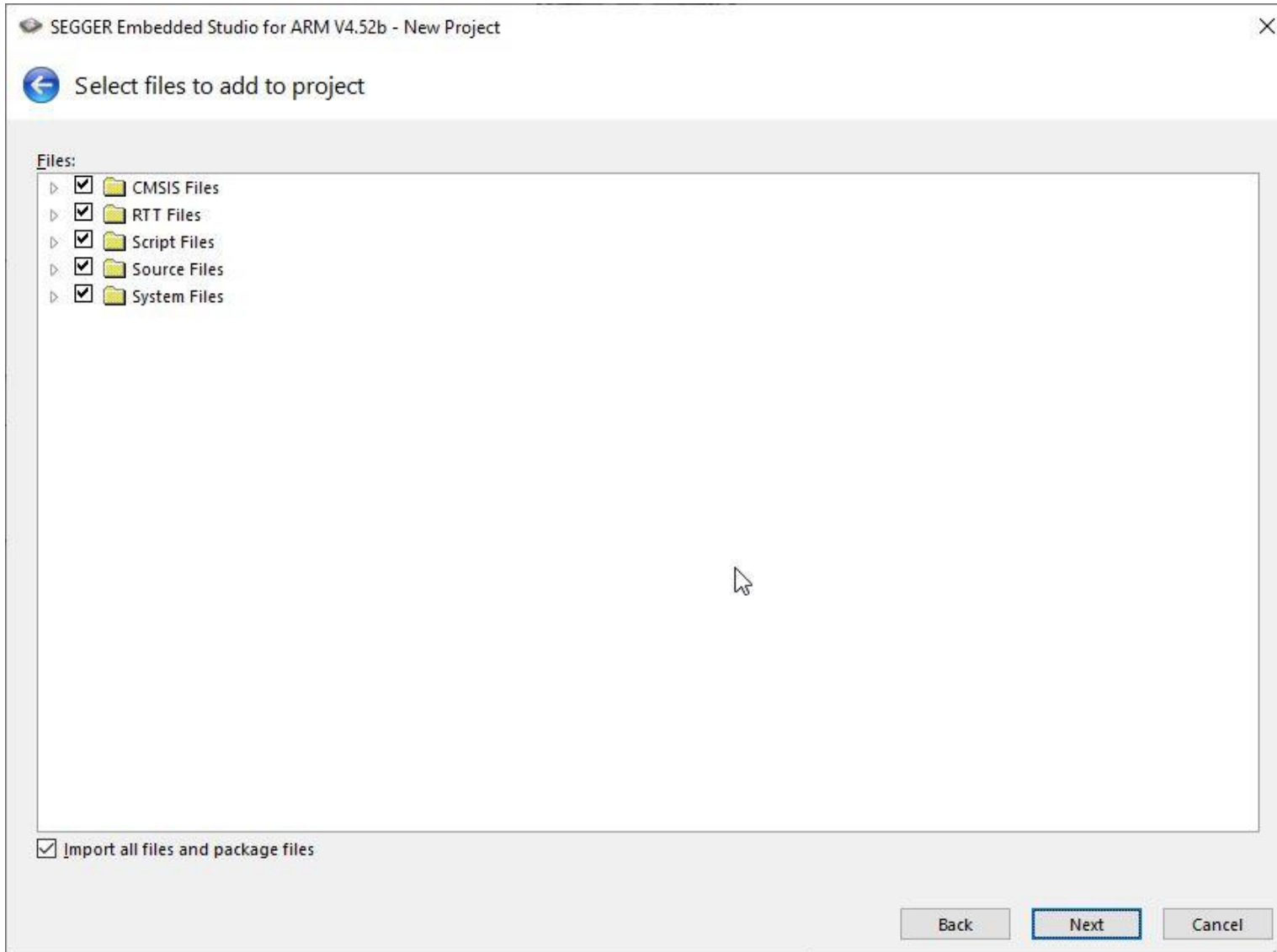
Embedded Studio Primer

MK20D7 Project Creation



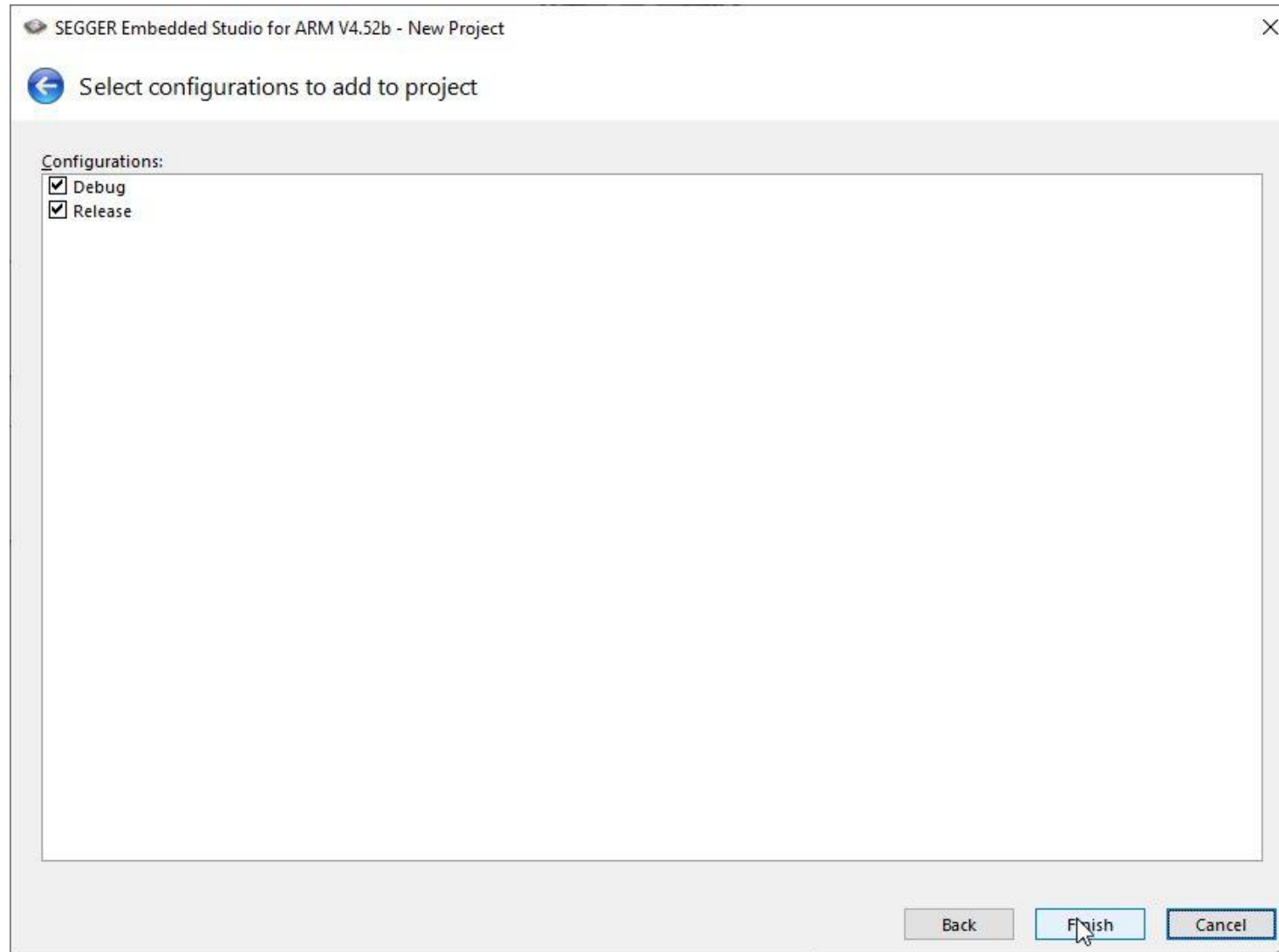
Embedded Studio Primer

MK20D7 Project Creation



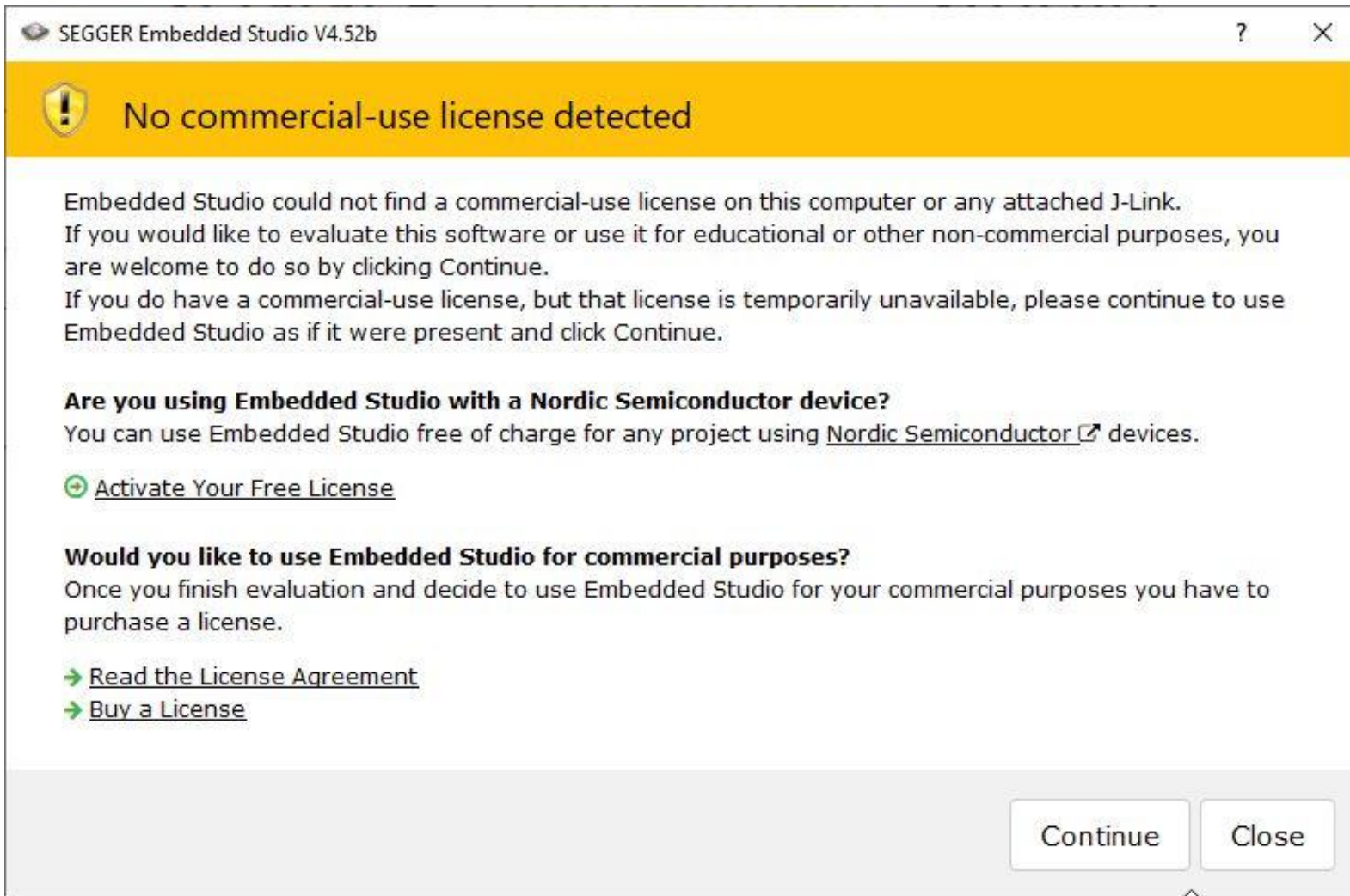
Embedded Studio Primer

MK20D7 Project Creation



Embedded Studio Primer

MK20D7 Project Creation



Embedded Studio Primer

MK20D7 Project Creation

TeensyProject - SEGGER Embedded Studio for ARM V4.52b (64-bit) - Non-Commercial License

File Edit View Search Navigate Project Build Debug Target Tools Window Help

Connect J-Link Ctrl+T, C
 Disconnect Ctrl+T, D
 Reconnect Ctrl+T, E
 Attach Debugger Ctrl+T, H
 Reset Ctrl+T, S
 Download TeensyProject Ctrl+T, L
 Verify TeensyProject Ctrl+T, V
 Erase All Ctrl+T, K
 Upload Range...
 Download File
 Verify File
 Start Cycle Counter
 Pause Cycle Counter
 Zero Cycle Counter Ctrl+T, Z
 Target Connection Properties

```

50 #include <stdio.h>
   #include <stdlib.h>
   /*****
   *
   *      main()
   *      Function description
   *      Application entry point.
   */
60 int main(void) {
   int i;

   for (i = 0; i < 100; i++) {
     printf("Hello World %d!\n", i);
   }
   do {
     i++;
   } while (1);
70 }
   /***** End of file *****/

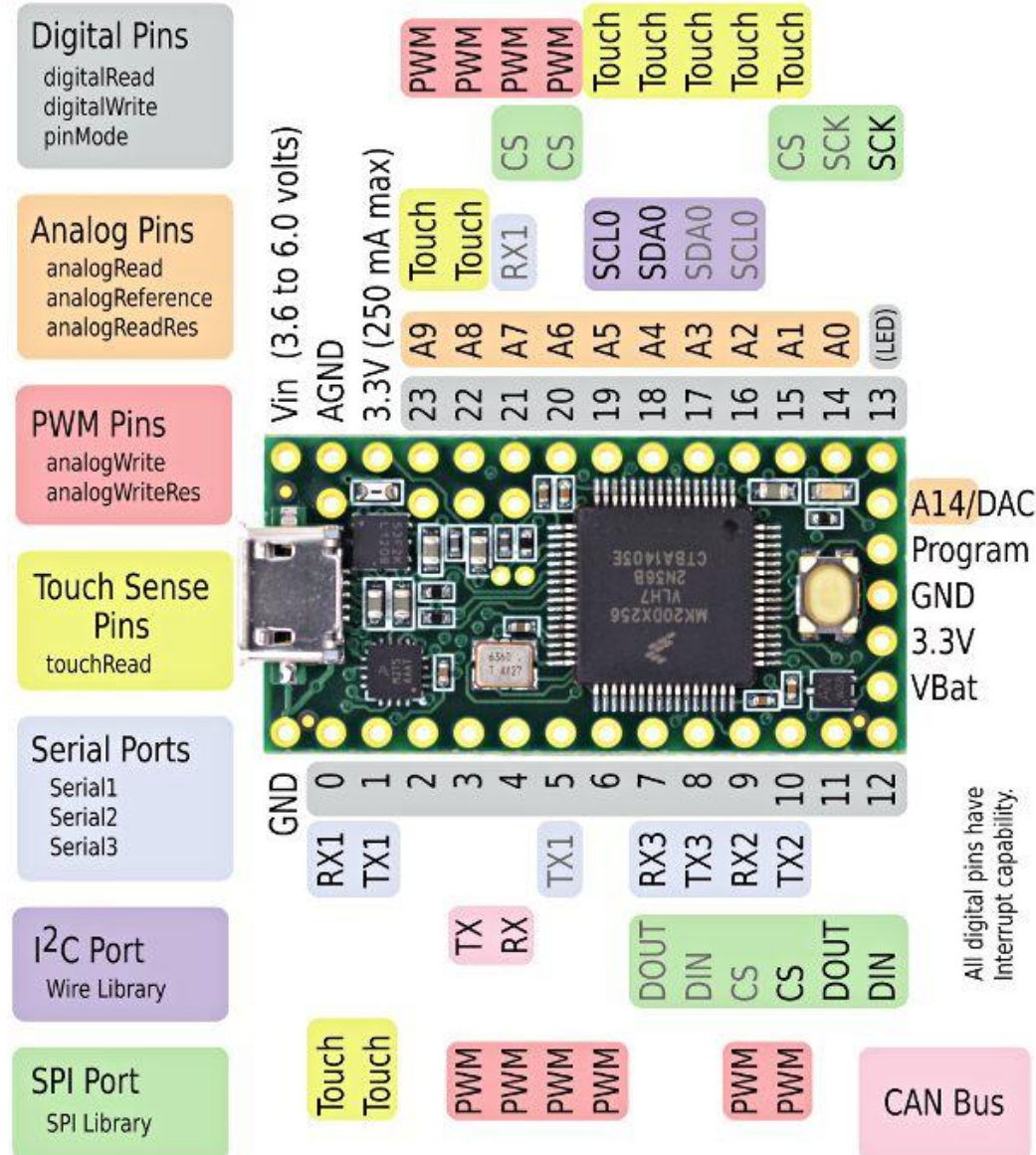
```

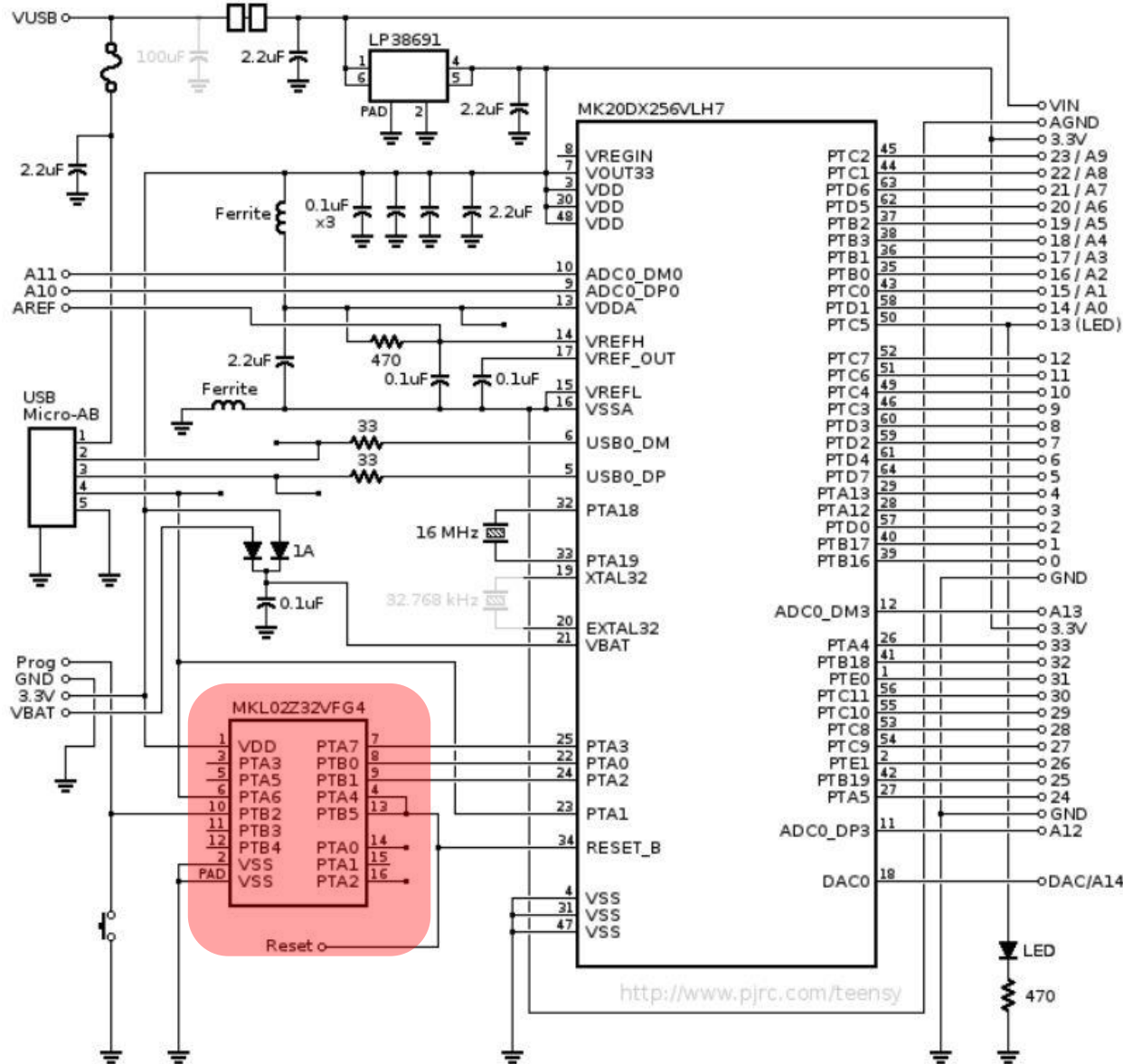
Show: Transcript Tasks

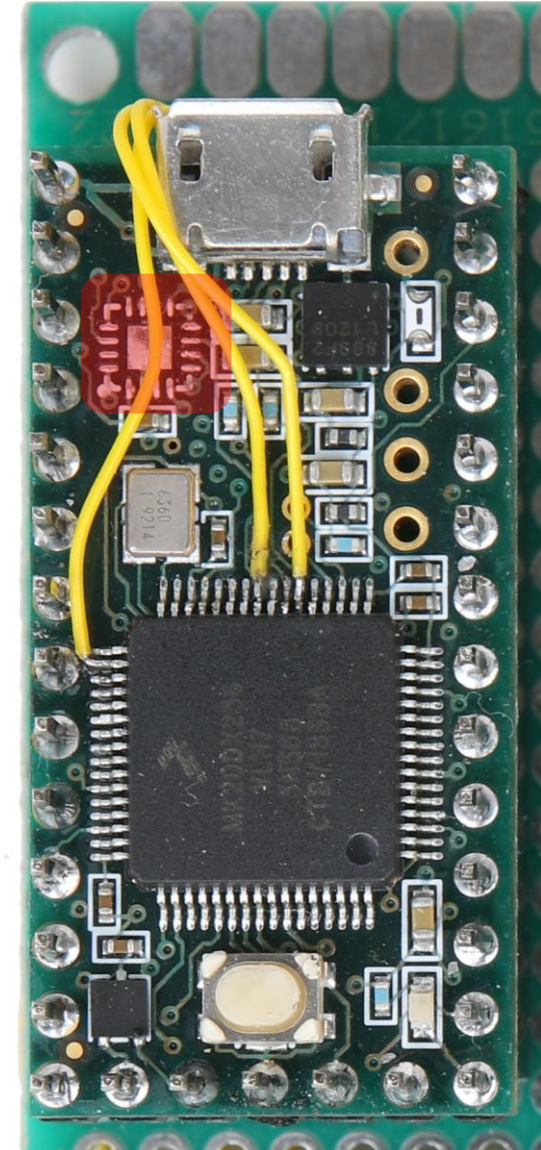
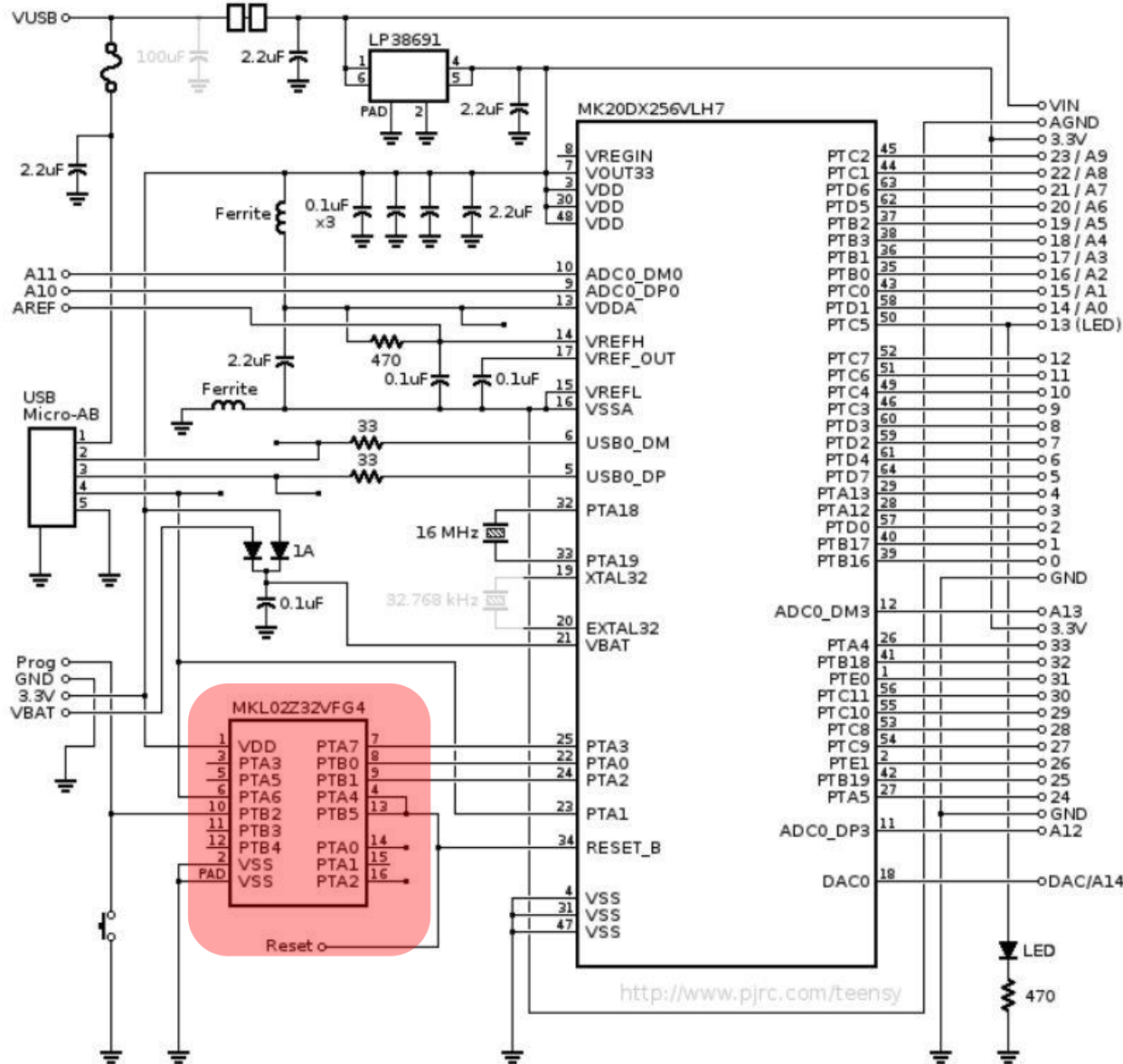
Connects to the selected target interface

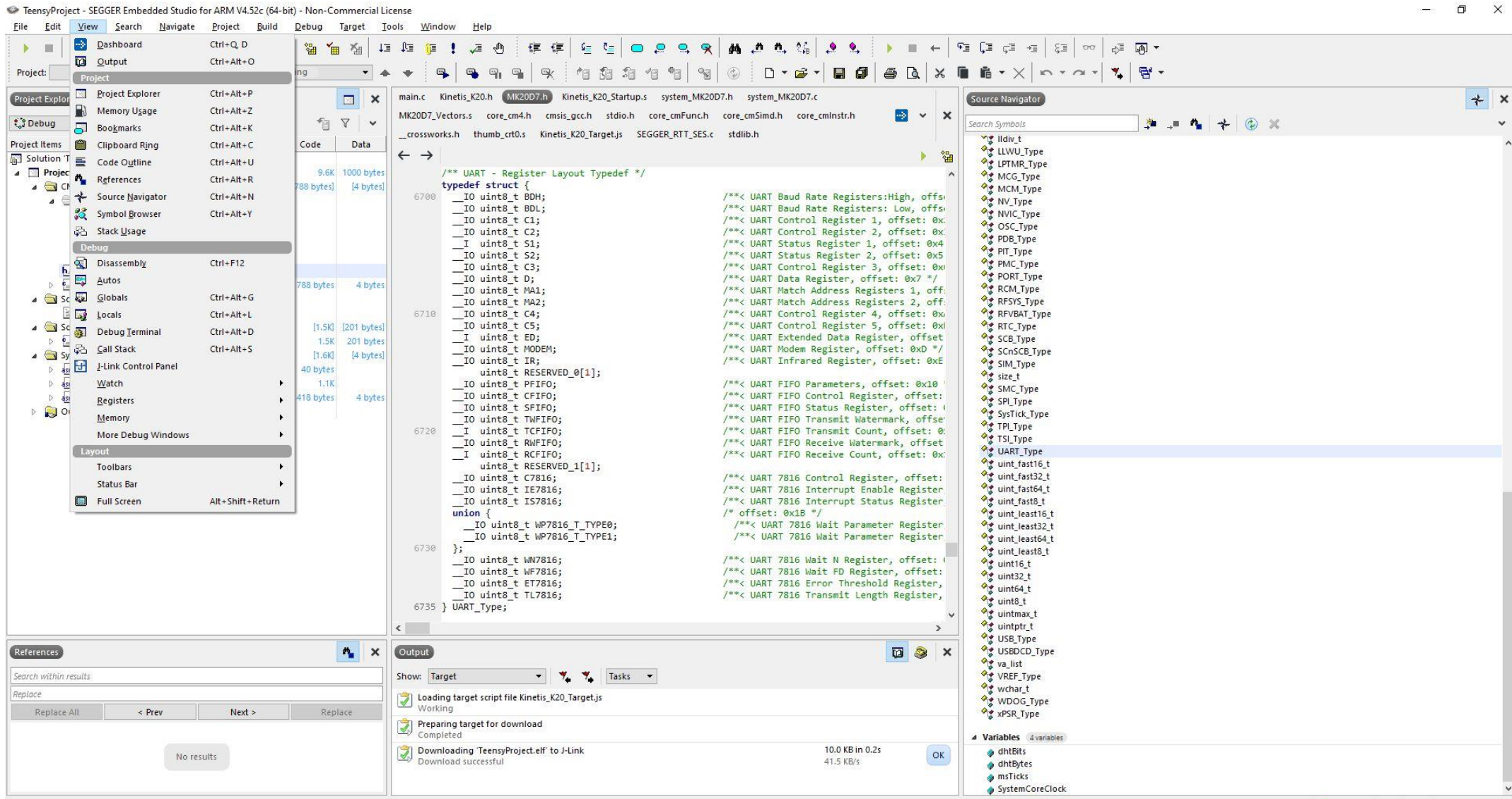
Disconnected (J-Link) Built OK INS (No editor)

Teensy 3.2 Hardware Modification









The screenshot displays the SEGGER Embedded Studio IDE for ARM V4.52c (64-bit). The interface includes a menu bar (File, Edit, View, Search, Navigate, Project, Build, Debug, Target, Tools, Window, Help), a toolbar, and several panels:

- Project Explorer:** Shows the project structure with folders for Project, Solution, and Source. A 'Debug' menu is open, listing options like Disassembly, Autos, Globals, Locals, Debug Terminal, Call Stack, J-Link Control Panel, Watch, Registers, Memory, and More Debug Windows.
- Code Editor:** Displays a C file named 'main.c' with a 'UART - Register Layout Typedef' structure. The code includes comments and defines various registers and control bits for the UART module.
- Source Navigator:** A tree view on the right showing a search for symbols and a list of available symbols, including 'UART_Type' which is currently selected.
- References:** A search box for finding references within the project.
- Output:** Shows the progress of downloading the target script file 'Kinetis_K20_Target.js' and preparing the target for download.
- Variables:** A list of variables defined in the current scope, including 'dhtBits', 'dhtBytes', 'msTicks', and 'SystemCoreClock'.

Embedded Studio Primer

Embedded Studio Fly-Over

TeensyProject - SEGGER Embedded Studio for ARM V4.52c (64-bit) - No

File Edit View Search Navigate Project Build Debug

Project Explorer

- Dashboard Ctrl+Q, D
- Output Ctrl+Alt+O
- Project Explorer Ctrl+Alt+P
- Memory Usage Ctrl+Alt+Z
- Bookmarks Ctrl+Alt+K
- Clipboard Ring Ctrl+Alt+C
- Code Outline Ctrl+Alt+U
- References Ctrl+Alt+R
- Source Navigator Ctrl+Alt+N
- Symbol Browser Ctrl+Alt+Y
- Stack Usage
- Disassembly Ctrl+F12
- Autos
- Globals Ctrl+Alt+G
- Locals Ctrl+Alt+L
- Debug Terminal Ctrl+Alt+D
- Call Stack Ctrl+Alt+S
- J-Link Control Panel
- Watch
- Registers
- Memory
- More Debug Windows
- Layout
 - Toolbars
 - Status Bar
 - Full Screen Alt+Shift+Return

license

Tools Window Help

main.c Kinetis_K20.h MK20D7.h Kinetis_K20_Startup.s system_MK20D7.h system_MK20D7.c

MK20D7_Vectors.s core_cm4.h cmsis_gcc.h studio.h core_cmFunc.h core_cmSimd.h core_cmInstr.h

__crossworks.h thumb_crt0.s Kinetis_K20_Target.js SEGGER_RTT_SES.c stdlib.h

```

/** UART - Register Layout Typedef */
typedef struct {
6700  __IO uint8_t BDH;
        __IO uint8_t BDL;
        __IO uint8_t C1;
        __IO uint8_t C2;
        __IO uint8_t S1;
        __IO uint8_t S2;
        __IO uint8_t C3;
        __IO uint8_t D;
        __IO uint8_t MA1;
        __IO uint8_t MA2;
6710  __IO uint8_t C4;
        __IO uint8_t C5;
        __IO uint8_t ED;
        __IO uint8_t MODEM;
        __IO uint8_t IR;
        uint8_t RESERVED_0[1];
        __IO uint8_t PFIPO;
        __IO uint8_t CFIFO;
        __IO uint8_t SFIFO;
        __IO uint8_t TWFIPO;
6720  __IO uint8_t TCFIFO;
        __IO uint8_t RWFIFO;
        __IO uint8_t RCFIFO;
        uint8_t RESERVED_1[1];
        __IO uint8_t C7816;
        __IO uint8_t IE7816;
        __IO uint8_t IS7816;
        union {
        __IO uint8_t WP7816_T_TYPE0;
        __IO uint8_t WP7816_T_TYPE1;
6730  };
        __IO uint8_t WN7816;
        __IO uint8_t WF7816;
        __IO uint8_t ET7816;
        __IO uint8_t TL7816;
6735  } UART_Type;
    
```

Source Navigator

Search Symbols

- lldiv_t
- LLWU_Type
- LPTMR_Type
- MCG_Type
- MCM_Type
- NV_Type
- NVIC_Type
- OSC_Type
- PDB_Type
- PIT_Type
- PMC_Type
- PORT_Type
- RCM_Type
- RFSYS_Type
- RVFBAT_Type
- RTC_Type
- SCB_Type
- SCnSCB_Type
- SIM_Type
- size_t
- SMC_Type
- SPI_Type
- SysTick_Type
- TPI_Type
- TSI_Type
- UART_Type
- uint_fast16_t
- uint_fast32_t
- uint_fast64_t
- uint_fast8_t
- uint_least16_t
- uint_least32_t
- uint_least64_t
- uint_least8_t
- uint16_t
- uint32_t
- uint64_t
- uint8_t
- uintmax_t
- uintptr_t
- USB_Type

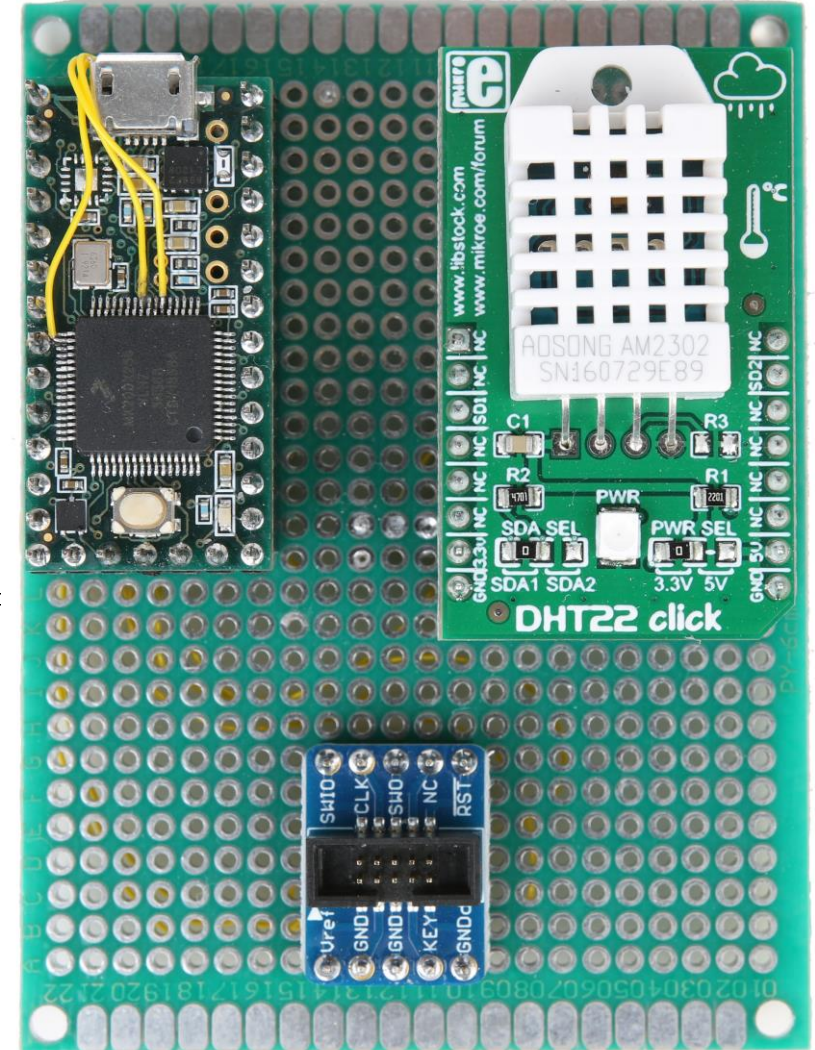
Embedded Studio Primer

Coding a DHT22 Application

```

//*****
/* FUNCTION PROTOTYPES
//*****
void delay_ms(uint32_t dlyTicks);
void uart0_putc(char c);
int uart0_printf(const char *fmt, ...);
uint8_t checkPulseWidth(unsigned int waitTime, unsigned int delta, bool pulseState);
char getDataBit(void);
uint8_t get40bits(void);
//*****
/* DEFINITIONS AND MACROS
//*****
#define LOW false
#define HIGH true
#define make16(varhigh,varlow) (((unsigned short)varhigh & 0xFF)* 0x100) + ((unsigned short)varlow & 0x00FF)
//*****
/* VARIABLES
//*****
uint32_t looper;
uint8_t scratch8;
volatile uint32_t msTicks = 0;
uint8_t bitIn;
uint8_t dhtBits[40] = {0};
uint8_t dhtBytes[5] = {0};
float tempC;
float tempF;
float humidity;

```



Embedded Studio Primer

Coding a DHT22 Application

```

//*****
//* PRINTF STUFF
//*****
typedef struct __printf_tag
{
    size_t charcount;
    size_t maxchars;
    char *string;
    int (*output_fn)(int, struct __printf_tag *ctx);
} __printf_t;

void uart0_putc(char c)
{
    while(!(UART0->S1 & UART_S1_TDRE_MASK));
    UART0->D = c;
}

int uart0_printf(const char *fmt, ...)
{
    int n;
    va_list ap;
    __printf_t iod;
    va_start(ap, fmt);
    iod.string = 0;
    iod.maxchars = INT_MAX;
    iod.output_fn = uart0_putc;
    n = __vfprintf(&iod, fmt, ap);
    va_end(ap);
    return n;
}

```

SEGGER Embedded Studio for ARM printf-style output

SEGGER Embedded Studio for ARM provides a solution for just this case by using some internal functions and data types in the SEGGER Embedded Studio for ARM library. These functions and types are define in the header file <__vfprintf.h>.

The first thing to introduce is the `__printf_t` type which captures the current state and parameters of the format conversion:

```

typedef struct __printf_tag
{
    size_t charcount;
    size_t maxchars;
    char *string;
    int (*output_fn)(int, struct __printf_tag *ctx);
} __printf_t;

```

This type is used by the library functions to direct what the formatting routines do with each character they need to output. If `string` is non-zero, the character is appended to the string pointed to by `string`; if `output_fn` is non-zero, the character is output through the function `output_fn` with the context passed as the second parameter.

The member `charcount` counts the number of characters currently output, and `maxchars` defines the maximum number of characters output by the formatting routine `__vfprintf`.

We can use this type and function to rewrite `uart0_printf`:

```

int uart0_printf(const char *fmt, ...)
{
    int n;
    va_list ap;
    __printf_t iod;
    va_start(ap, fmt);
    iod.string = 0;
    iod.maxchars = INT_MAX;
    iod.output_fn = uart0_putc;
    n = __vfprintf(&iod, fmt, ap);
    va_end(ap);
    return n;
}

```

Embedded Studio Primer

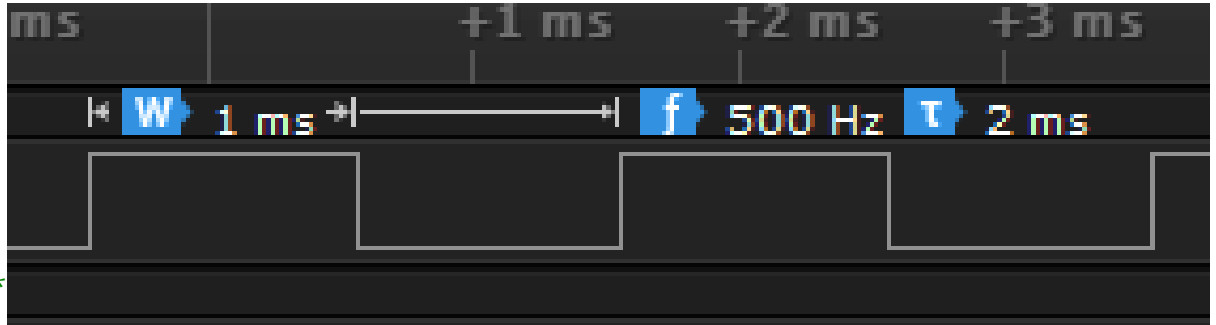
Coding a DHT22 Application

```

//*****
//* DELAY MICROSECONDS
//*****
void delay_us(uint32_t dlyus)
{
    FTM0->SC = FTM_SC_CLKS(0) | FTM_SC_PS(0); //stop uS counter
    FTM0->CNTIN = 0;
    FTM0->SC = FTM_SC_CLKS(1) | FTM_SC_PS(0); //start uS counter
    FTM0->CNT = 0;
    while(FTM0->CNT < ((dlyus - 1) * 36));
}
//*****
//* DELAY MILLISECONDS
//*****
void delay_ms(uint32_t dlyTicks)
{
    uint32_t curTicks;

    curTicks = msTicks;
    while ((msTicks - curTicks) < dlyTicks);
}
//*****
//* SYSTICK TIMER INTERRUPT HANDLER
//*****
void SysTick_Handler(void)
{
    msTicks++;
}

```



Embedded Studio Primer

Coding a DHT22 Application

```

//*****
/* MAIN
//*****
int main(void)
{
  uint8_t i,j;
  init();
  do{
    if(startPulse())
    {
      if(get40bits())
      {
        bitIn = 0;
        for(i=0; i<5; i++) // i is the BYTE counter
        {
          for(j=0; j<8; j++) // j gives the current position of a bit in i'th BYTE
          {
            if( dhtBits[ 8*i + j ] ) //5 bytes(8*i) of 8 bits(j) each = 40 bits
            {
              bitIn |= (1<<(7-j)); //shift in 8 bits to make a byte in dhtBytes array
            }
            //shift in 7,6,5...0 - 7-j is bit counter
          }
          //note that we only shift in a 1 and skip zeroes
          dhtBytes[i] = bitIn;
          bitIn = 0;
        }

        humidity = make16(dhtBytes[0],dhtBytes[1]);
        tempC = make16(dhtBytes[2],dhtBytes[3]);
        tempC /= 10;
        tempF = (tempC * 1.8) + 32;
        humidity /= 10;
        uart0_printf("Temp=%3.2fF, Temp=%3.2fC, Humidity=%3.2f%% \r\n",tempF,tempC,humidity);
      }
    }
  }while(1);
}

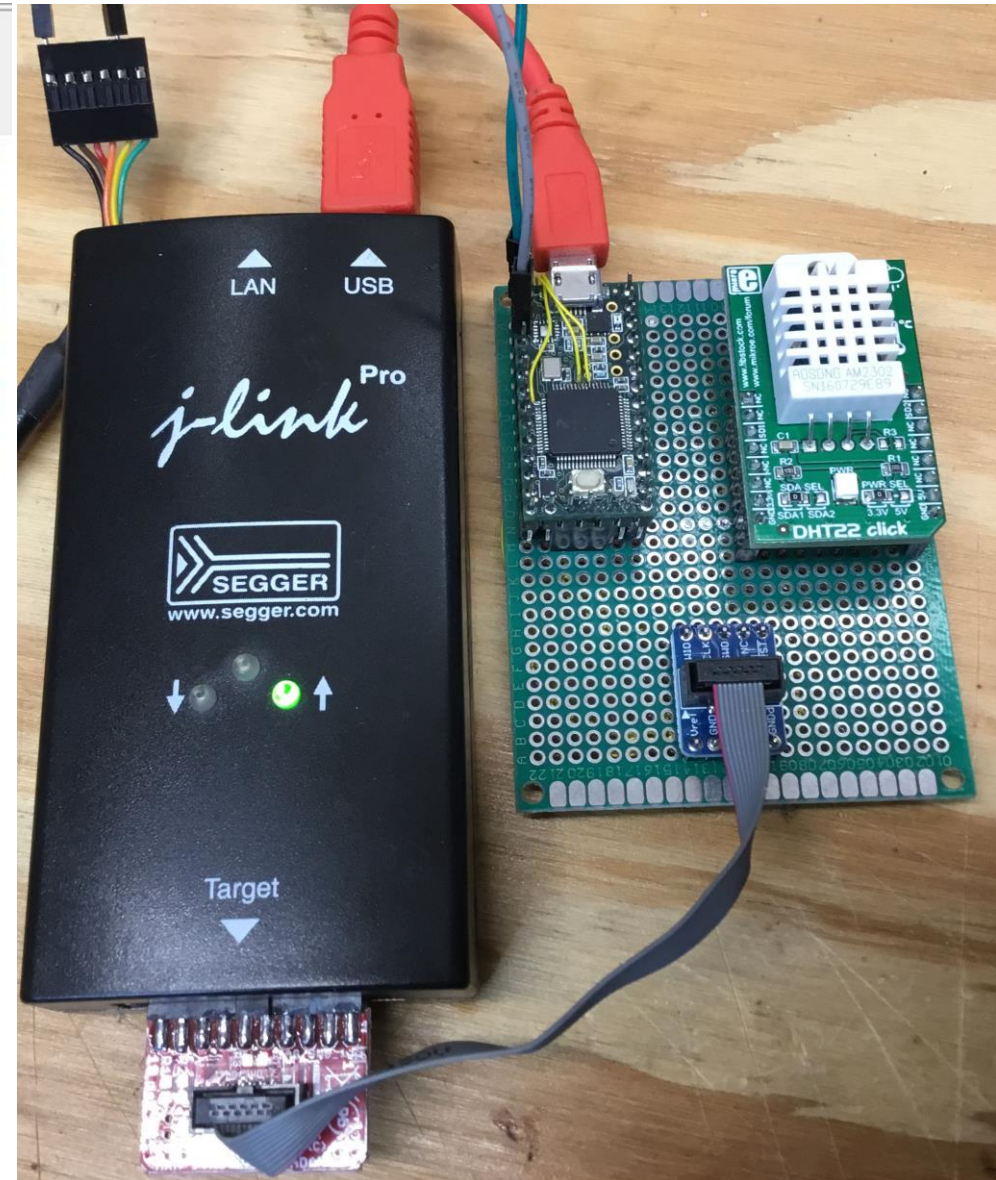
```

Terminal Emulator

```

Temp=78.08F, Temp=25.60C, Humidity=58.70%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.20%
Temp=78.08F, Temp=25.60C, Humidity=58.10%

```



Thank you for attending

Please consider the resources below:

- <https://www.mikroe.com>
- <https://www.pjrc.com>
- <https://www.saleae.com>
- <https://www.nxp.com>
- <https://www.segger.com>



Thank You

Sponsored by

