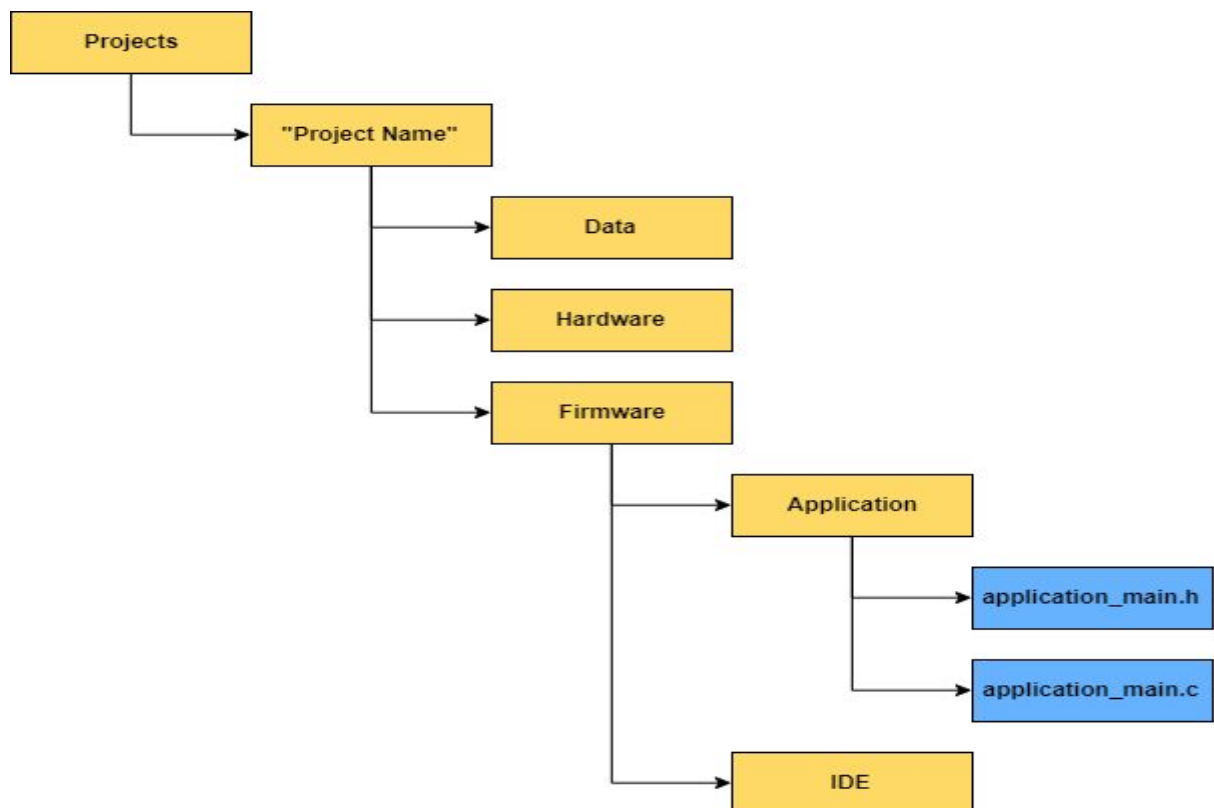


# Tutorial para a criação de um projeto base para o treinamento de “Desenvolvimento de Software Driver para Periféricos de Microcontroladores.”

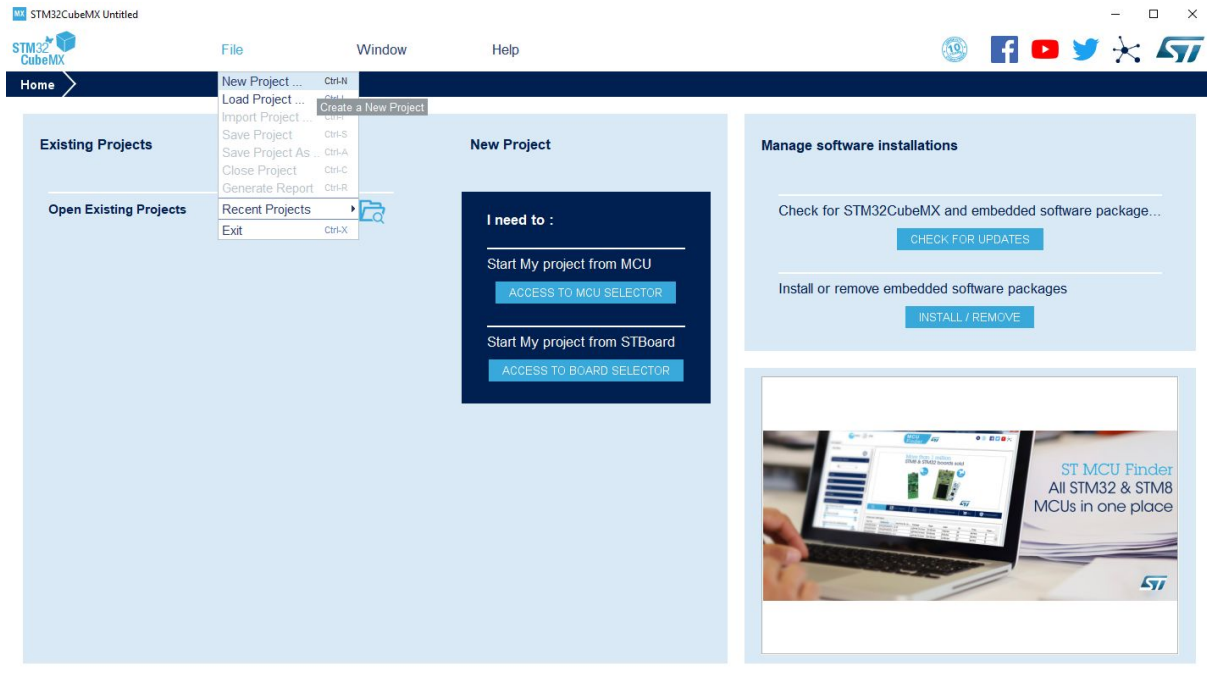
**1º** - Criar uma árvore de diretórios com a estrutura igual ao da **imagem 1**, sendo necessário apenas a criação dos diretórios (em amarelo) sendo que os arquivos “application\_main.c” e “application\_main.h” (em azul) serão criados posteriormente através da IDE.

Nesta árvore de diretórios:

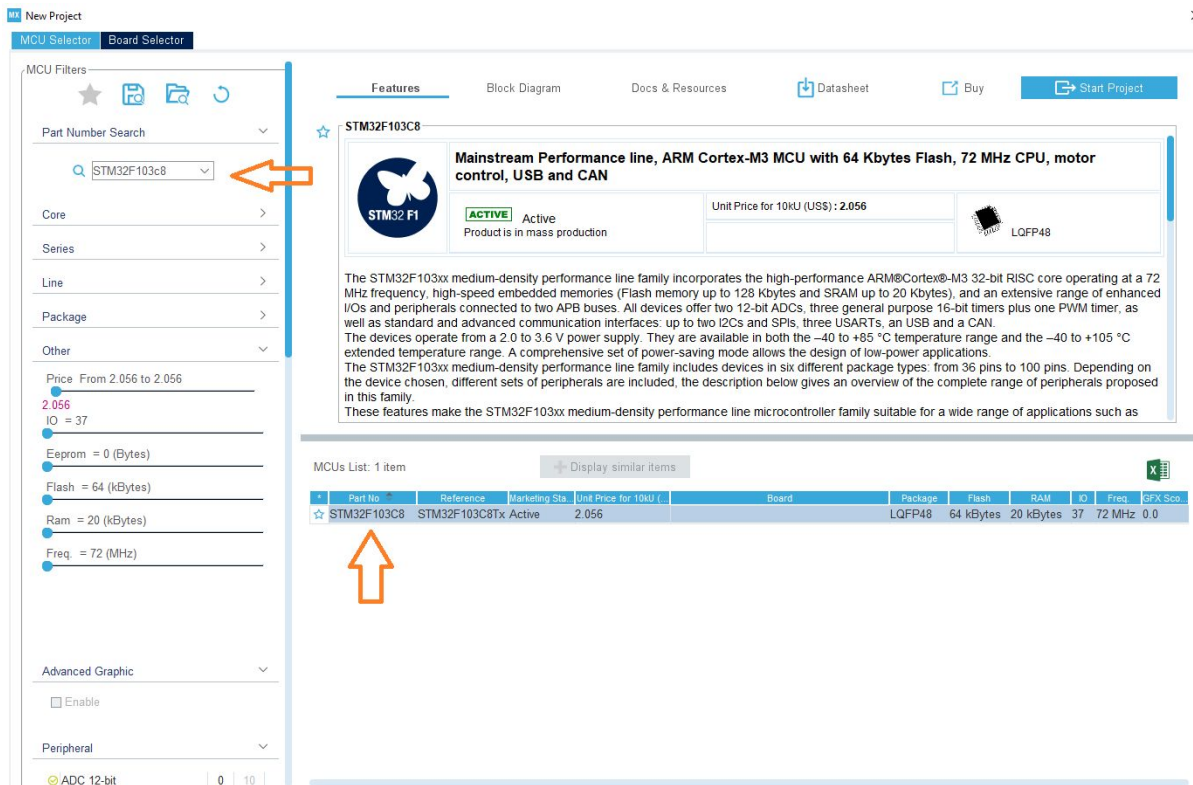
- **Project**: Contém diretórios com os projetos desenvolvidos.
- **“Project Name”**: Nome de um projeto em desenvolvimento.
- **Data**: Contém documentos, fotos, planilhas, etc com dados referentes ao projeto.
- **Hardware**: Diretório com esquemáticos, layout, e outros arquivos ligados ao desenvolvimento do hardware do projeto.
- **Firmware**: Contém o código fonte do projeto e os arquivos referentes ao projeto gerado pela IDE.
- **Application**: Possui todo o código fonte desenvolvido.
- **IDE**: Possui arquivos referentes ao projeto criados pela IDE.



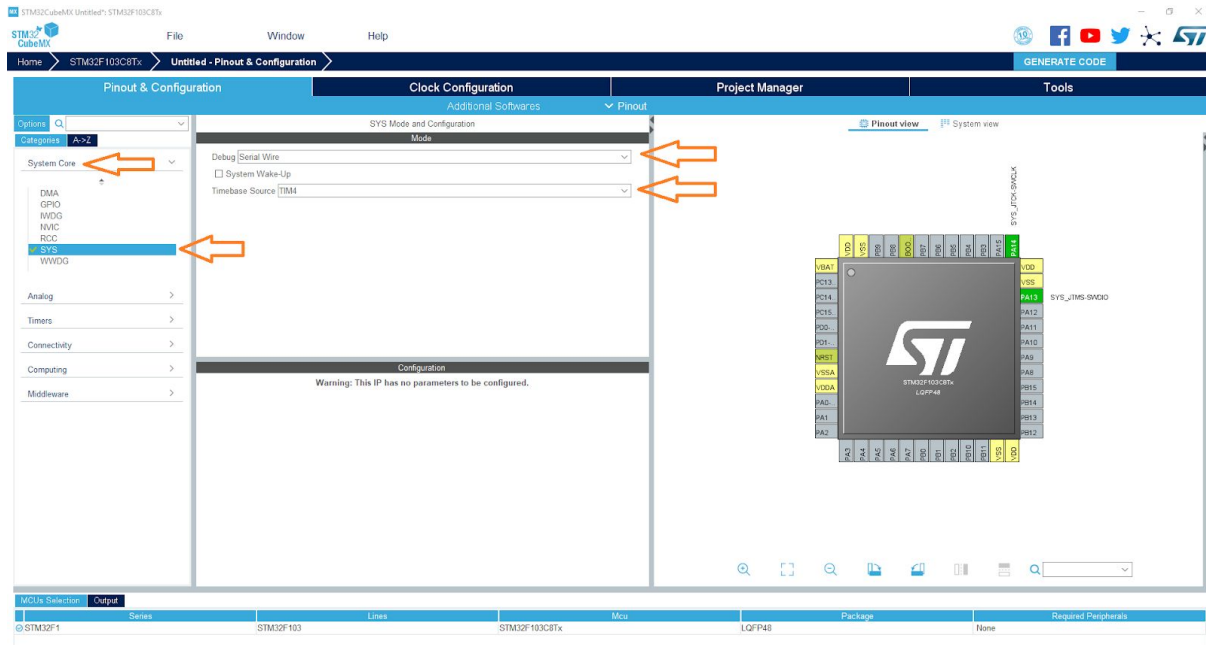
2º - Abra o programa STM32CubeMX (a versão nas imagens é a V5.1.0), abra a aba “File” e seleciona a opção “New Project”:



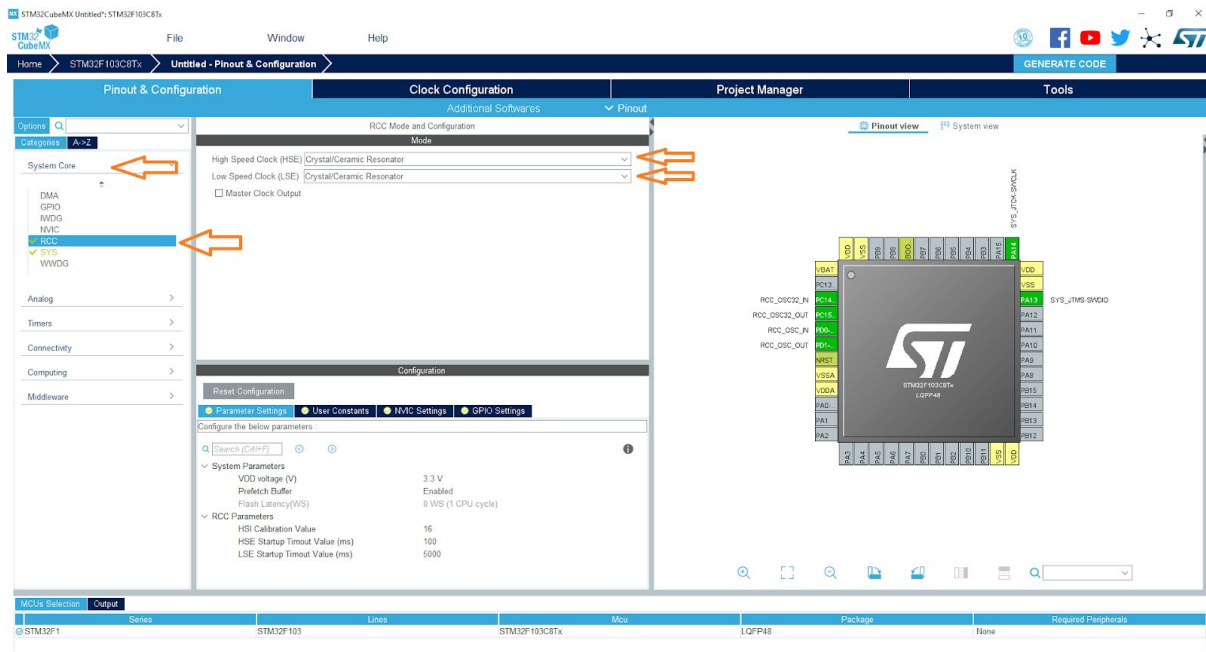
3º - Na tela que se abrirá, no campo “Part Number Search” insira o código do microcontrolador “STM32F103C8” e em “MCUs List” dê dois cliques sobre o elemento que possuir a descrição do microcontrolador utilizado:



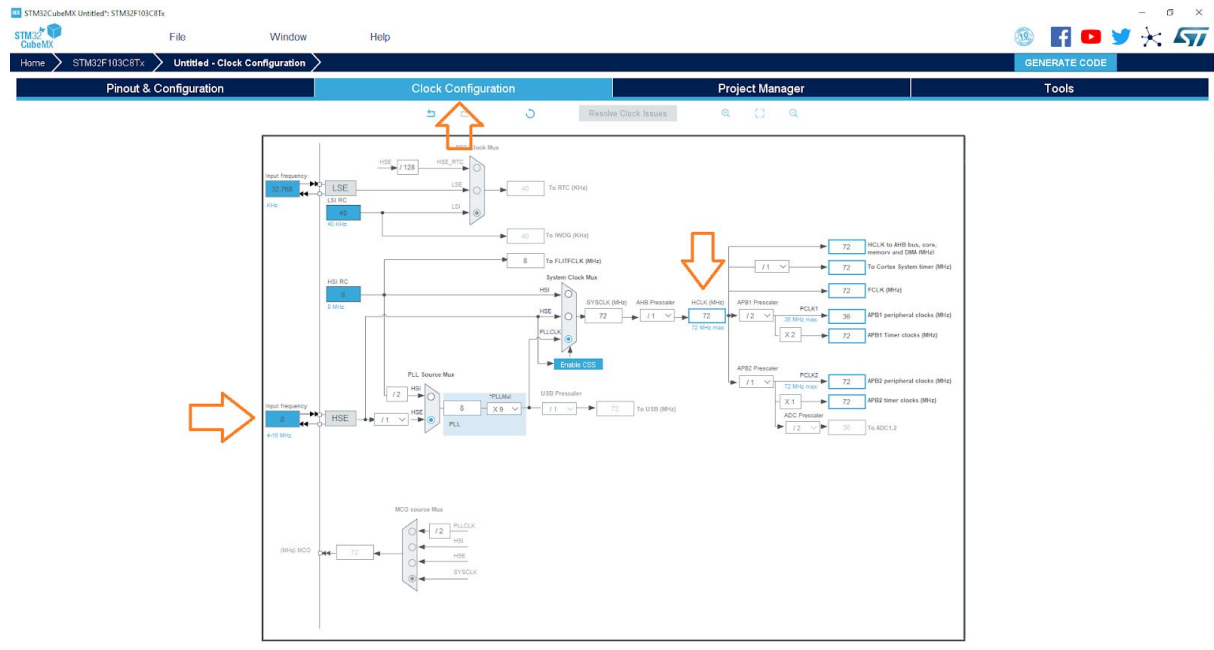
4º - Uma nova janela se abrirá com a aba “**Pinout & Configuration**” aberta, e em “**System Core**” selecione a opção “**SYS**” e, no menu que se abrirá, para “**Debug**” selecione “**Serial Write**” e para “**Timebase Source**” selecione “**TIM4**”:



5º - Agora em “**System Core**” selecione a opção “**RCC**” e, na janela que se abrirá, em “**High Speed Clock (HSE)**” selecione “**Crystal/Ceramic Resonator**” e em “**Low Speed Clock (LSE)**” selecione “**Crystal/Ceramic Resonator**”:



**6º - Abra a aba “Clock Configuration” e em “Input Frequency” tenha certeza de que este campo possui o valor “8” e em “HCLK (MHz)” insira o valor “72”:**

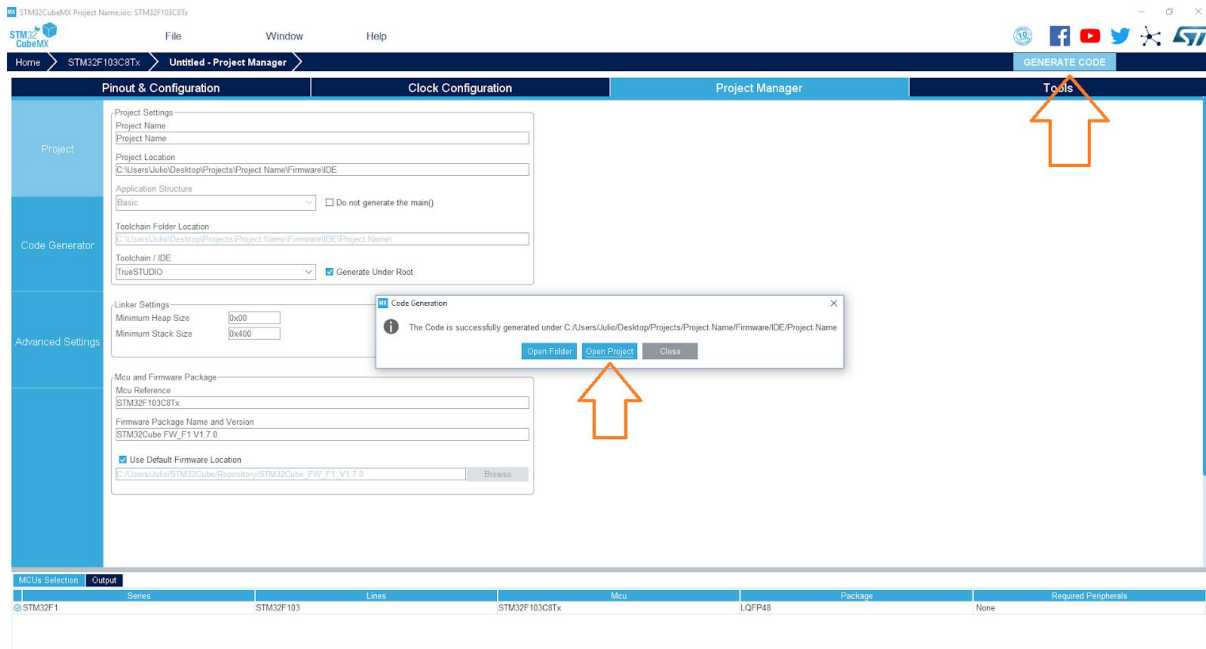


**7º - Selecione a aba “Project Manager” e selecione “Project” no canto superior direito. Em “Project Name” insira o nome do projeto e em “Project Location” seleciona o diretório “IDE” da nossa árvore de diretórios, informando o caminho completo através do botão “Browse”. Em “Toolchain / IDE” selecione “TrueSTUDIO” e em “Minimum Heap Size” insira o valor “0x00”:**

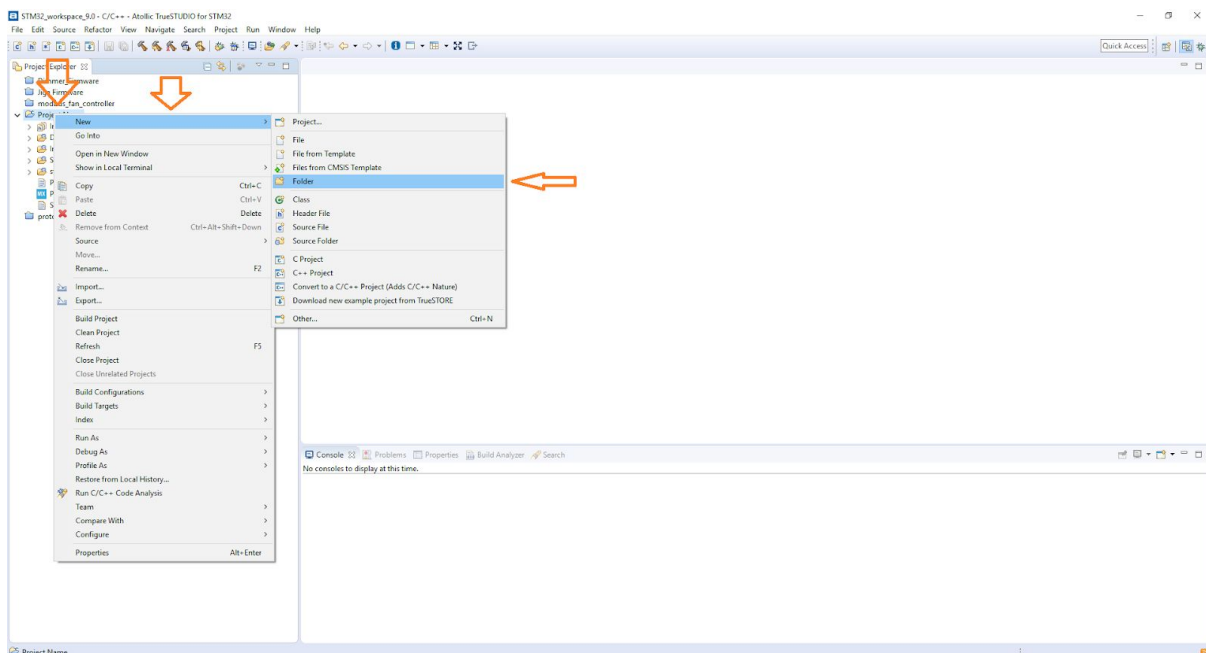
The screenshot shows the STM32CubeMX Project Manager tab. The 'Project' tab is selected. The 'Project Name' is set to 'STM32F103C8Tx'. The 'Project Location' is set to 'C:\Users\Usuario\Desktop\Projects\Project Name\Firmware\IDE'. The 'Toolchain / IDE' is set to 'TrueSTUDIO'. The 'Minimum Heap Size' is set to '0x00'. The 'Minimum Stack Size' is set to '0x400'. The 'MCU and Firmware Package' section shows 'STM32F103C8Tx' and 'STM32Cube\_FW\_F1\_V1.7.0'. Orange arrows point to the 'Project' tab, 'Project Name', 'Project Location', 'Toolchain / IDE', and 'Minimum Heap Size' fields.

MCU's Selection	Series	Lines	Mcu	Package	Required Peripherals
STM32F1	STM32F103	STM32F103C8Tx	STM32F103C8Tx	LQFP48	None

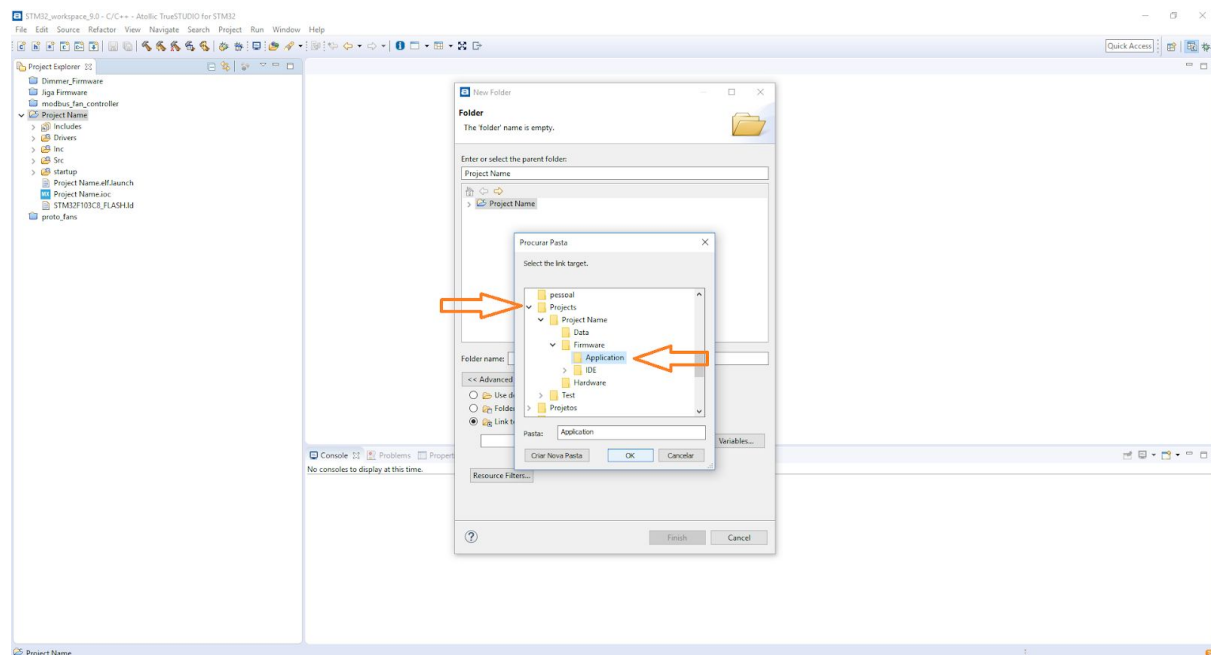
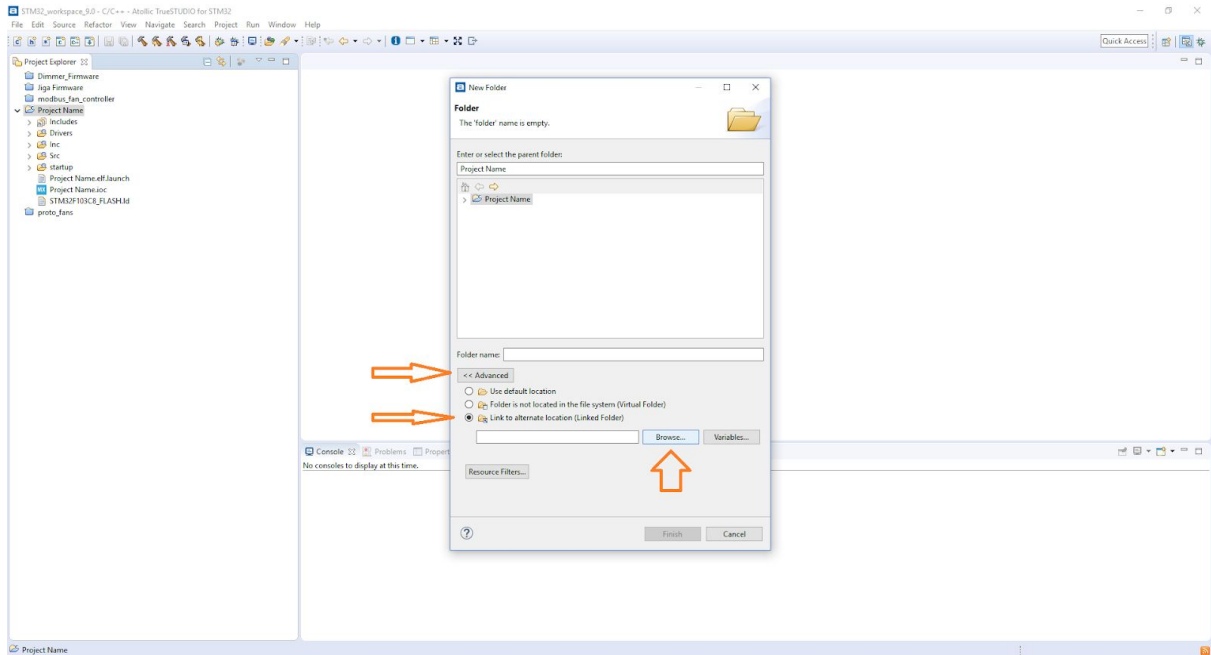
8º - Finalmente, clique sobre o botão **“GENERATE CODE”** no canto superior direito e, quando surgir a notificação de ação, selecione a opção **“Open Project”**:



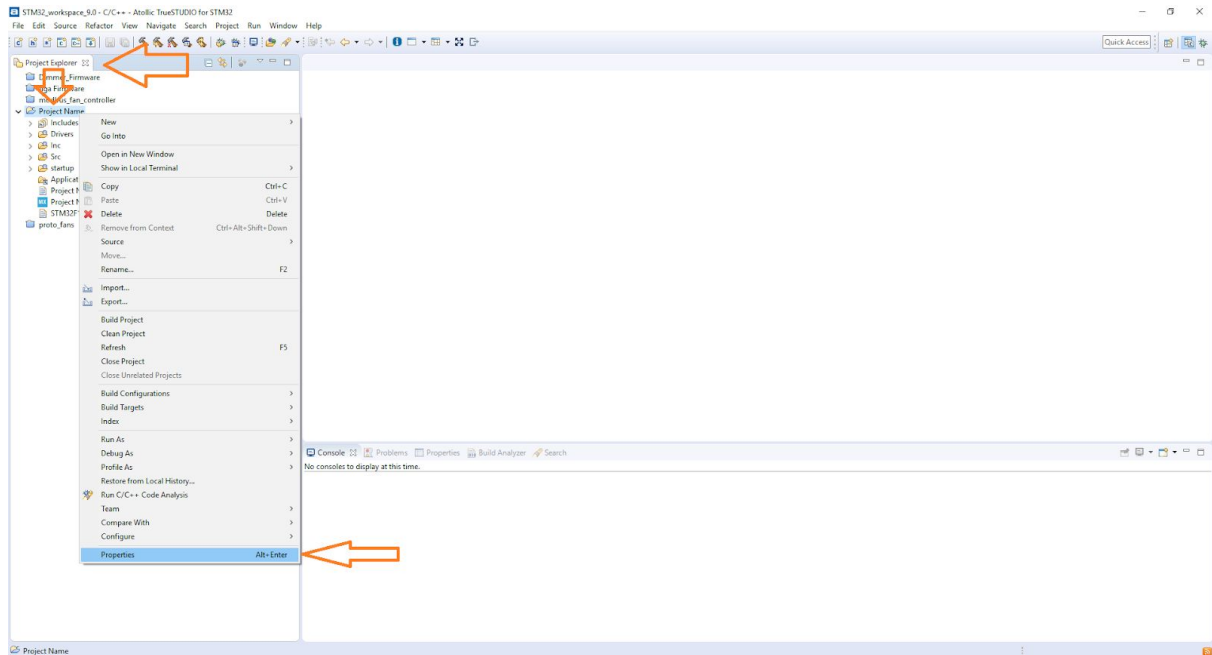
9º - A IDE **“Atollic TrueSTUDIO for STM32”** será aberta com o projeto desenvolvido pelo software **“CubeMX”**, na janela **“Project Explorer”** clique com o botão direito sobre o diretório do projeto, surgindo um menu, e selecione a opção **“New”** e selecione a opção **“Folder”**:



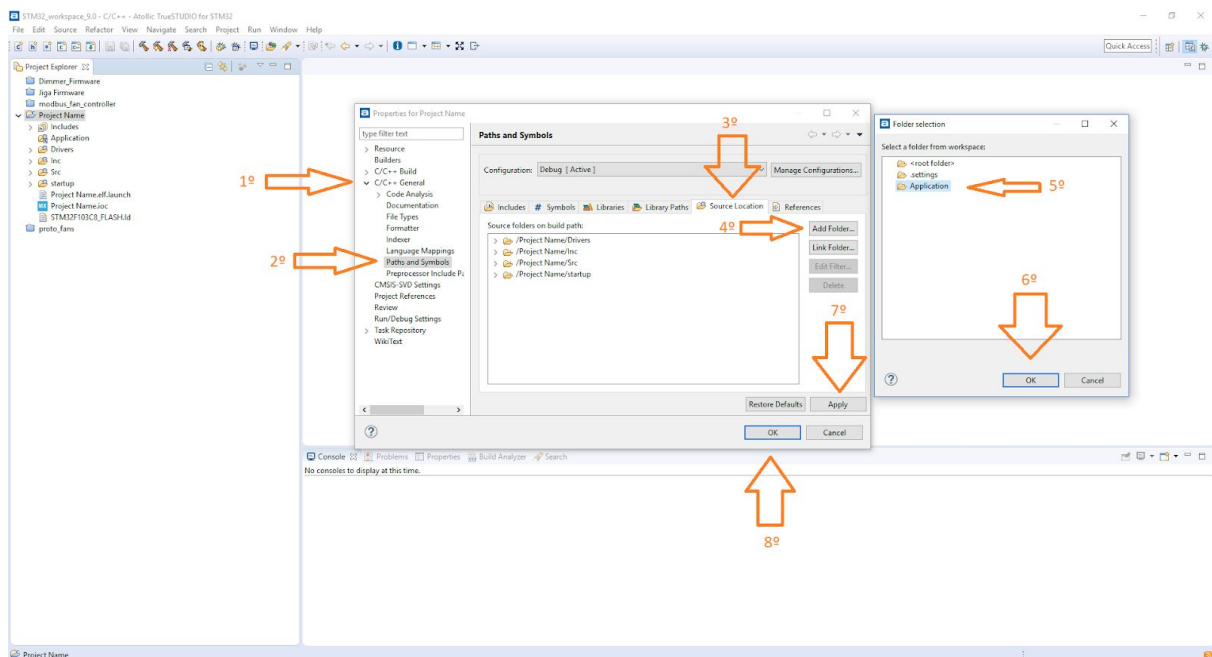
**10º** - Uma janela com o nome “**New Folder**” se abrirá e nela clique sobre o botão “**Advanced**” e selecione a opção “**Link to alternate location (Linked Folder)**” e, em seguida, clique no botão “**Browse...**” informando o caminho completo até o diretório “**Application**” que compõe a árvore de diretórios criada no passo 1:



**11º** - Na janela “**Project Explorer**” clique com o botão direito sobre a pasta do projeto e, no menu que surgirá, selecione a opção “**Properties**”:

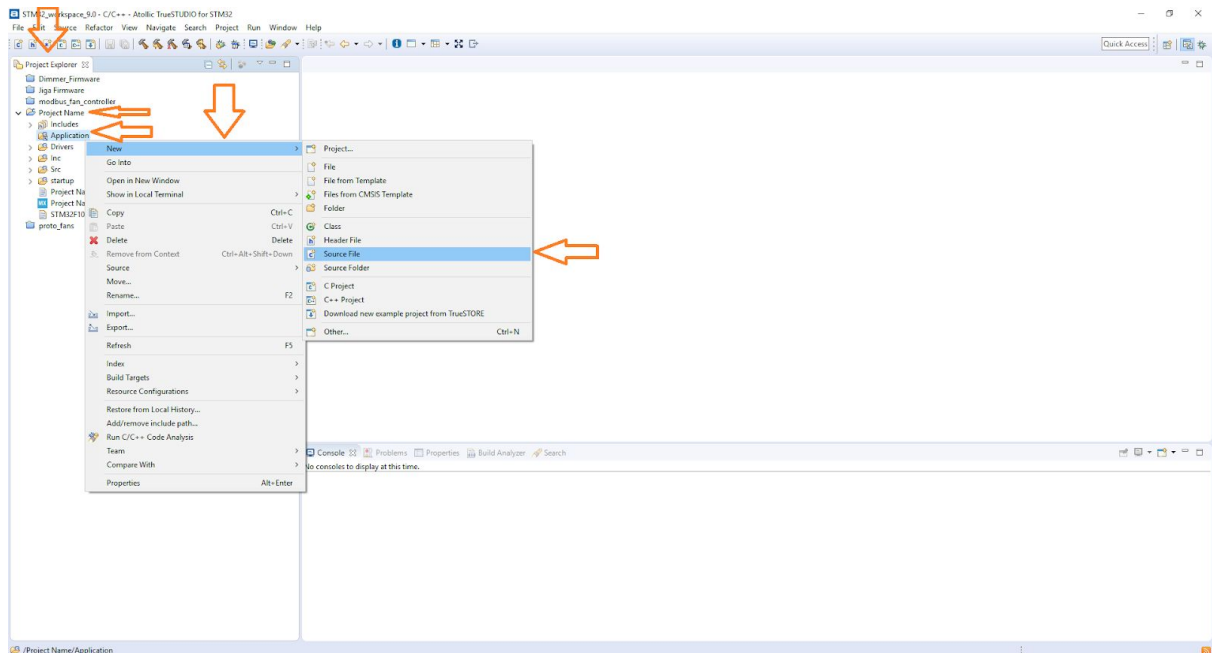


**12º** - Na janela que se abrirá (“**Properties for \*nome do seu projeto\***”), selecione na aba lateral esquerda “**C/C++ General**” e selecione a opção “**Path and Symbols**”, selecione a aba “**Source Location**” e clique sobre o botão “**Add Folder...**”. Na janela que se abrirá (“**Folder selection**”) selecione o diretório “**Application**” e clique sobre o botão “**OK**”. Em seguida clique sobre o botão “**Apply**” e “**OK**” na janela “**Properties for \*Nome Seu do Projeto\***”:

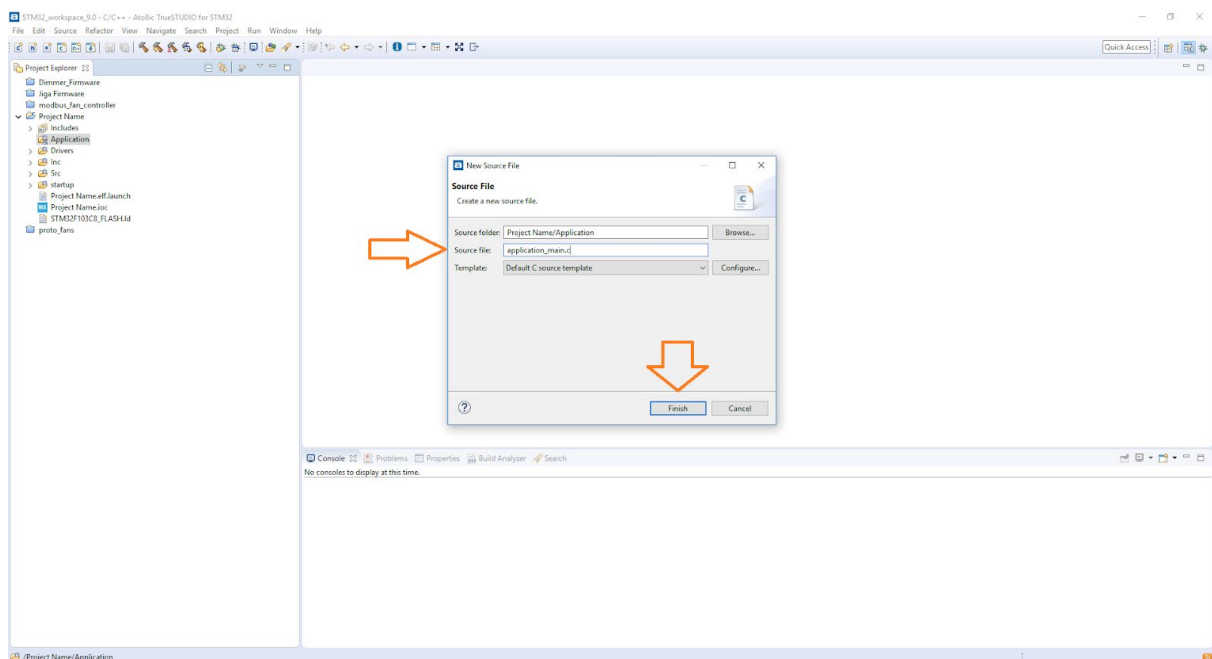




**13º** - Na janela “**Project Explorer**” clique com o botão direito sobre a pasta “**Application**”, que está dentro da pasta com o nome do seu projeto, e selecione em “**New**” a opção “**Source File**”:



**14º** - Na janela que se abrirá (“**New Source File**”), em “**Source File**”, insira o nome “**application\_main.c**” e clique no botão “**Finish**”:





15º - Faça novamente o passo 13 e em vez de selecionar a opção “**Source File**” selecione “**Header File**” e, na janela que se abrirá com o nome “**New Header File**”, em “**Header File**”, insira o nome “**application\_main.h**” e clique no botão “**Finish**”.

16º - No arquivo “application\_main.c” insira o seguinte código:

```
#include "application_main.h"

void app_run(void)
{
    while(1){

    }
}
```

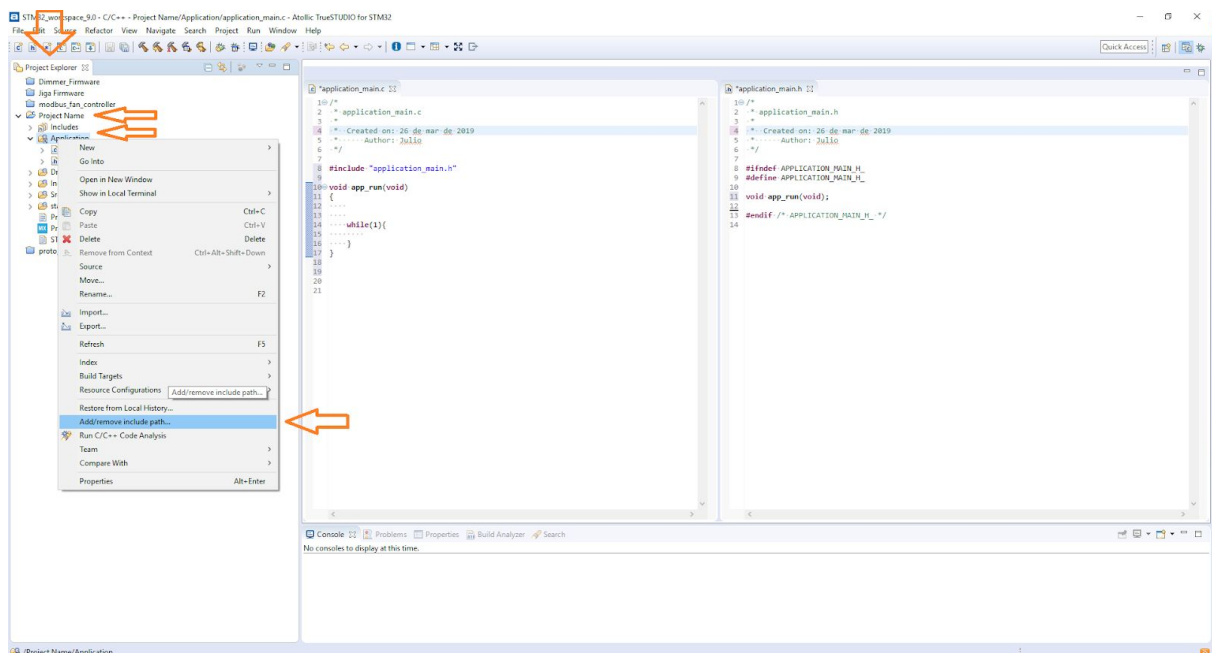
17º - No arquivo “application\_main.h” insira o seguinte código:

```
#ifndef APPLICATION_MAIN_H_
#define APPLICATION_MAIN_H_

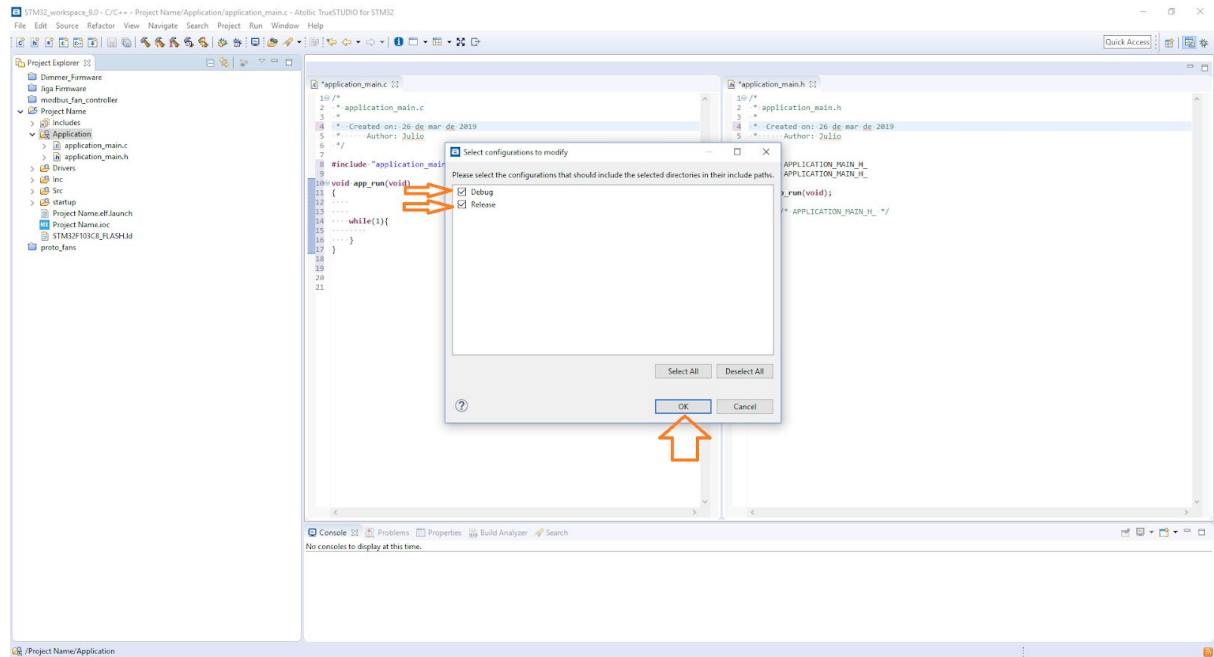
void app_run(void);

#endif
```

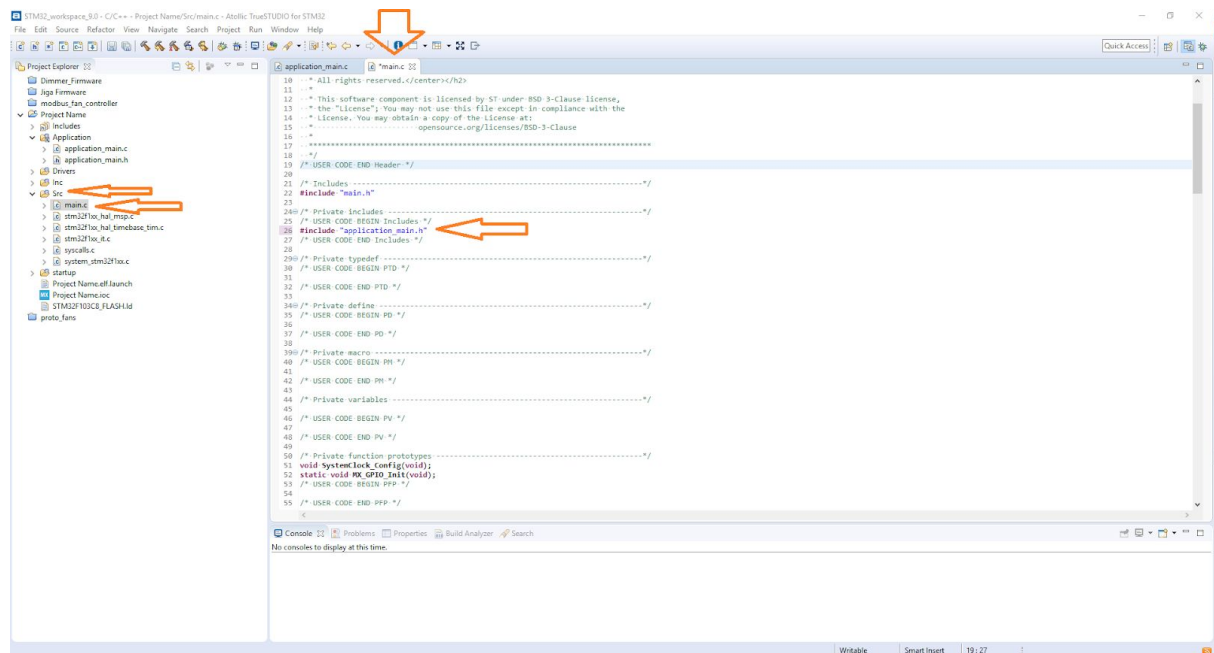
18º - Na janela “**Project Explorer**”, clique com o botão direito em “**Application**” e selecione a opção “**Add/remove include path...**”:

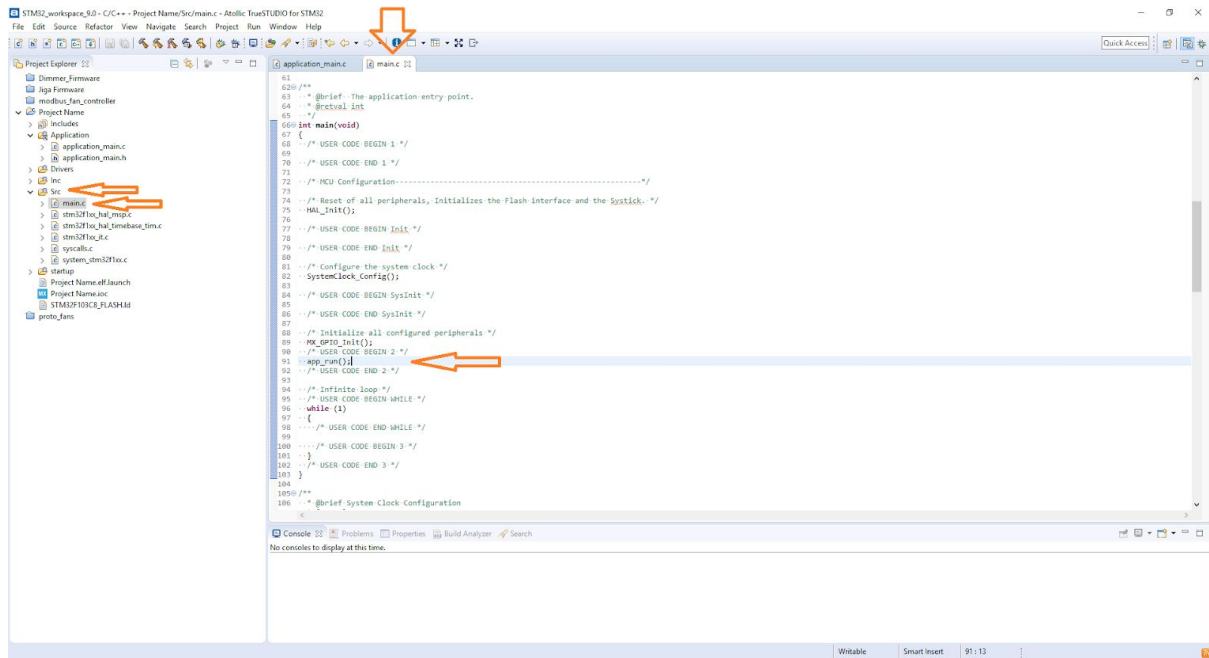


**19º -** Selecione as opções **“Debug”** e **“Release”** na janela que surgirá com o nome de **“Select configurations to modify”** e depois clique no botão **“OK”**:

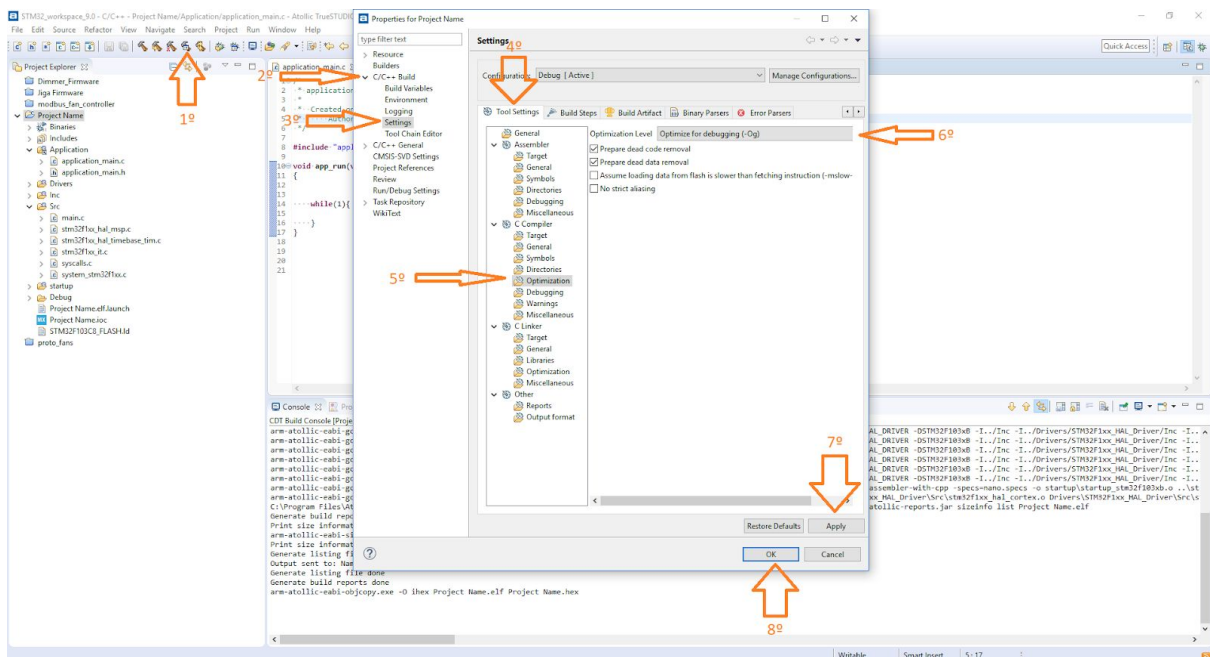


**20º -** Na janela **“Project Explorer”**, em **“Src”** abra o arquivo **“main.c”** e entre as linhas de código **“/\* USER CODE BEGIN Includes \*/** e **“/\* USER CODE END Includes \*/** insira o código **“#include “application\_main.h”**, e entre as linhas **“/\* USER CODE BEGIN 2 \*/** e **“/\* USER CODE END 2 \*/** insira o código **“app\_run();”**:





**21º** - No canto superior esquerdo, clique no símbolo de um martelo com uma pequena caixa texto azul e a janela “**Properties for \*nome do seu projeto\***” surgirá. Abra a opção “**C/C++ Build**” e selecione “**Settings**”, em seguida selecione “**Tool Settings**” e em “**C Compiler**” selecione “**Optimization**” e na opção “**Optimization Level**” selecione “**Optimize for debugging (-Og)**”. Clique no botão “**Apply**” e em seguida no botão “**OK**”:



**21º** - No canto superior esquerdo, clique no símbolo de um martelo para compilar o programa. Na Janela “**Console**” aparecerá, ao final da compilação, a mensagem “**Build Finished**” caso toda a operação de configuração tenha sido feita corretamente.

