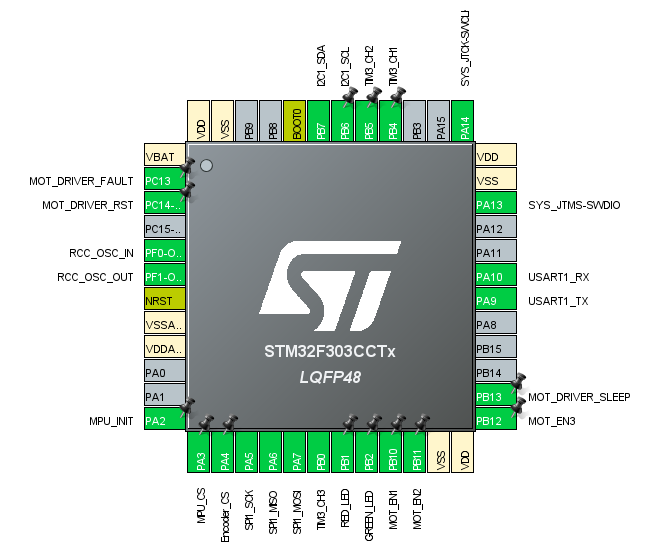
Mikrovaldiklis STM32F303CCT7, pakeistas iš STM32F303C8T6, nes neužteko 64 kB Flash atminties

SPI Encoder‘s ir MPU9250 ant tos pačios magistralės nėra galimas. Reikia keisti į I2C, nes kitos SPI nėra. Yra keli variantai:

1. Atsisakyti MPU9250
2. Naudoti MPU9250 kartu su INA226 ir EEPROM
3. Atsisakyti INA226 o vietoje jos nuolatos naudoti MPU9250

Pasirinkta naudoti atskyra 12C2 magistralę

Mikrovaldiklio konfiguravimas



1. RCC -> High Speed Clock - > Crystal/Ceramic Resonator
2. SYS - > Debug -> Serial Wire
3. SYS -> TimeBase Source -> TIM1
4. SYS -> TimeBase Source -> TIM4
   1. Clock Source -> Internal CLock
   2. Prescaler -> 72
   3. Counter Period -> 1
5. Timer -> TIM3 ->
   1. Clock Source -> Internal CLock
   2. Channel 1 -> PWM Generation CH1
   3. Channel 2 -> PWM Generation CH2
   4. Channel 3 -> PWM Generation CH3
   5. Prescaler -> 3
   6. Counter Period -> 1000
6. USART1 -> Baud Rate -> 115200
7. SPI1 (SPI Mode 1; Standart is Mode 3)
   1. Data Size -> 16 bit
   2. Prescaler -> 64 (1.125 MBits/s)
   3. Clock Polarity (CPOL) -> Low
   4. Clock Phase (CPHA) -> 2 Edge
   5. DMA Settings -> Add -> SPI1\_RX ------> Priority -> Medium
   6. DMA Settings -> Add -> SPI1\_TX ------> Priority -> Medium
   7. NVIC Settings -> SPI global Interrupt -> Enable Box
8. I2C1
   1. I2C -> I2C
   2. DMA Settings -> Add -> I2C1\_RX ------> Priority -> Medium
   3. DMA Settings -> Add -> I2C 1\_TX ------> Priority -> Medium
   4. NVIC Settings -> I2C1 event global Interrupt -> Enable Box
   5. NVIC Settings -> I2C1 error interrupt -> Enable Box
9. Middleware -> FreeRTOS -> CMSIS\_V1
10. Clock Configuration
    1. HSE -> 16 MHz
    2. Enable HSE round Box
    3. HCLK (MHz) -> 72 MHz
    4. Enter ENTER for recalculatio
11. Generate project files

**Empy Project Size**

text data bss dec hex filename

26224 24 6280 32528 7f10 GMM\_BIG\_V2.elf

**Flash usage**

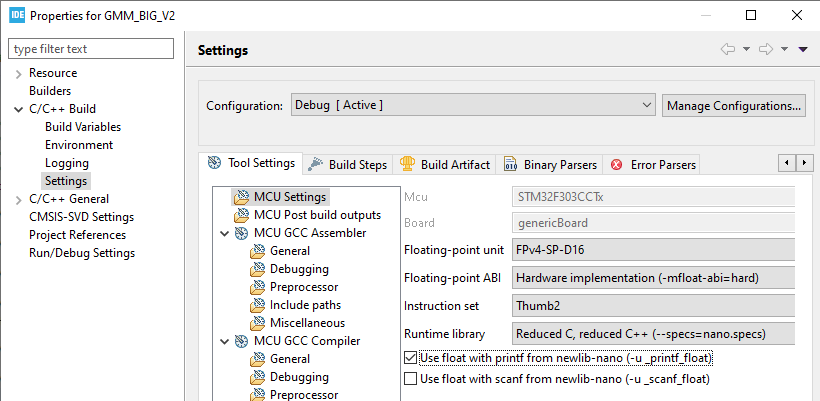
Text + data

**RAM usage**

Data + bss

**Projekto parametru konfiguravimas STM32CubeIDE aplinkoje, kad veiktu paprastas printf.**

Project properties -> C/C++ Build -> Settings -> MCU Settings -> Enable „Use float with printf from newlin-nano“



Padeda panaikinti viena warninga, bet užima daugiau vietos atmintyje

MCU GCC Compiler -> Preprocessor -> Add „configUSE\_NEWLIB\_REENTRANT“

MCU G++Compiler -> Preprocessor -> Add „configUSE\_NEWLIB\_REENTRANT“

Reikia specializuotos bibliotekos newlib.c (projekto source kataloge yra). Ši biblioteka pakeičia FreeRTOS naudojama heap4.c ir sysmem.c, kur ištaiso STM paliktas klaidas. Atitinkmai šitus failus šaliname iš kompiliavimo, nes rodis rimtas klaidas dėl dubliavimosi. Pažymime failą dešniu palytės klavišu ir pasirenkame Resource configuration ir nuspaudžiame Exclude from build. Atsidariusiame lange uždedame varneles ir spaudžiame OK. Atliekame su abejais failais

Paprasto printf kodas

#ifdef \_\_GNUC\_\_

#define PUTCHAR\_PROTOTYPE int \_\_io\_putchar(int ch)

#else

#define PUTCHAR\_PROTOTYPE int fputc(int ch, FILE \*f)

#endif /\* \_\_GNUC\_\_ \*/

/\*\*

\* @brief Retargets the C library printf function to the USART.

\* @param None

\* @retval None

\*/

PUTCHAR\_PROTOTYPE

{

/\* Place your implementation of fputc here \*/

/\* e.g. write a character to the USART2 and Loop until the end of transmission \*/

HAL\_UART\_Transmit(&huart1, (uint8\_t \*)&ch, 1, 0xFFFF);

return ch;

}

**C++ atvejis**

Pakeitus main.c galunę į cpp atsiras prenešimas „ISO C++ forbids converting ...“.

Atidarome cmsis\_os.h ir surandame eilutę

281 **char** \*name; ///< Thread name

Papildome į

281 **const** **char** \*name; ///< Thread name

Įvertiname primtf funkciją, kadangi ji nėra C++, o C

**#ifdef** \_\_cplusplus

**extern** "C" {

**#endif**

PUTCHAR\_PROTOTYPE

{

/\* Place your implementation of fputc here \*/

/\* e.g. write a character to the USART2 and Loop until the end of transmission \*/

HAL\_UART\_Transmit(&huart1, (uint8\_t \*)&ch, 1, 0xFFFF);

**return** ch;

}

**#ifdef** \_\_cplusplus

}

**#endif**

Gijos dysis 128 yra permažas sėkmingam printf datbui. Uždėjus 256 jau veikia.

broadcast(uavcan.thirdparty.msg.MotorCommand(cmd=[4,-1,0]))