**Monitoring Anti-Tamper System**

**Project Goal:**

* Application of monitoring anti-tamper reflective sensor

**Interface for Project:**

* C / C++

**Installing Prerequisites**

* **STM32CubeIDE** – install from [here](https://www.st.com/en/development-tools/stm32cubeide.html)
* Git : Get the latest git from [here](https://git-scm.com/downloads)

**Hardware:**

* **STM32L486xx**: Reference link from [here](https://www.st.com/resource/en/datasheet/stm32l486jg.pdf)
* **KazuardBoard**: Custom board
* **Core**: Arm® 32-bit Cortex®-M4 CPU with FPU1.
* **Power supply**: 71 V to 3.6 V
* **VDD**: 1.71V to 1.9V

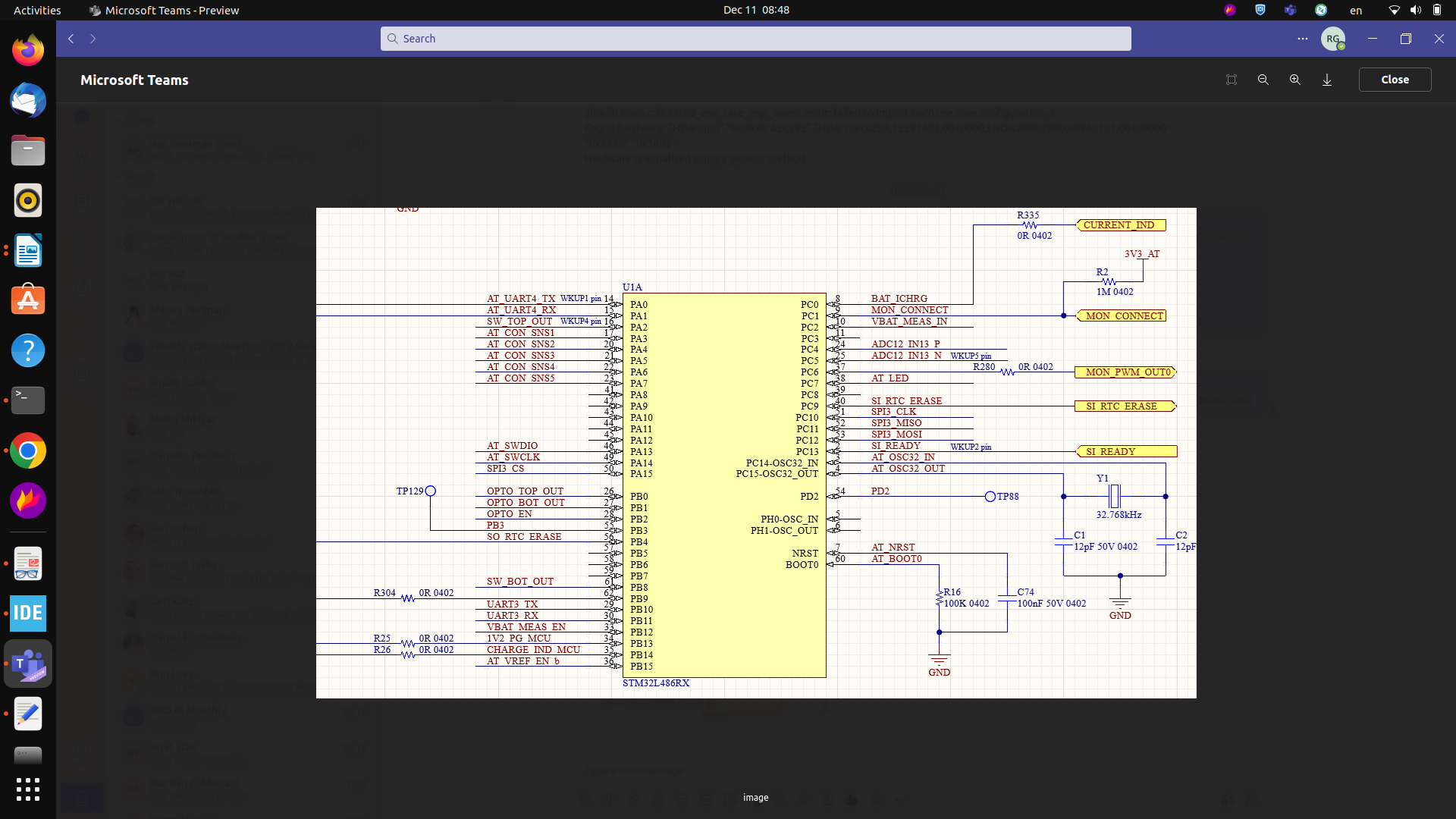
**Commponnent:**

* **STM32L486xx -** U1
* **Reflective Sensor (QRE1113)** top side – U34
* **Switch Sensor buttom side** – SW2
* **Switch Sensor top side** – SW1
* **Rtc Secure-input** – U15
* **Rtc Secure-output** – U20
* **Red Led** – LED for AT top side..

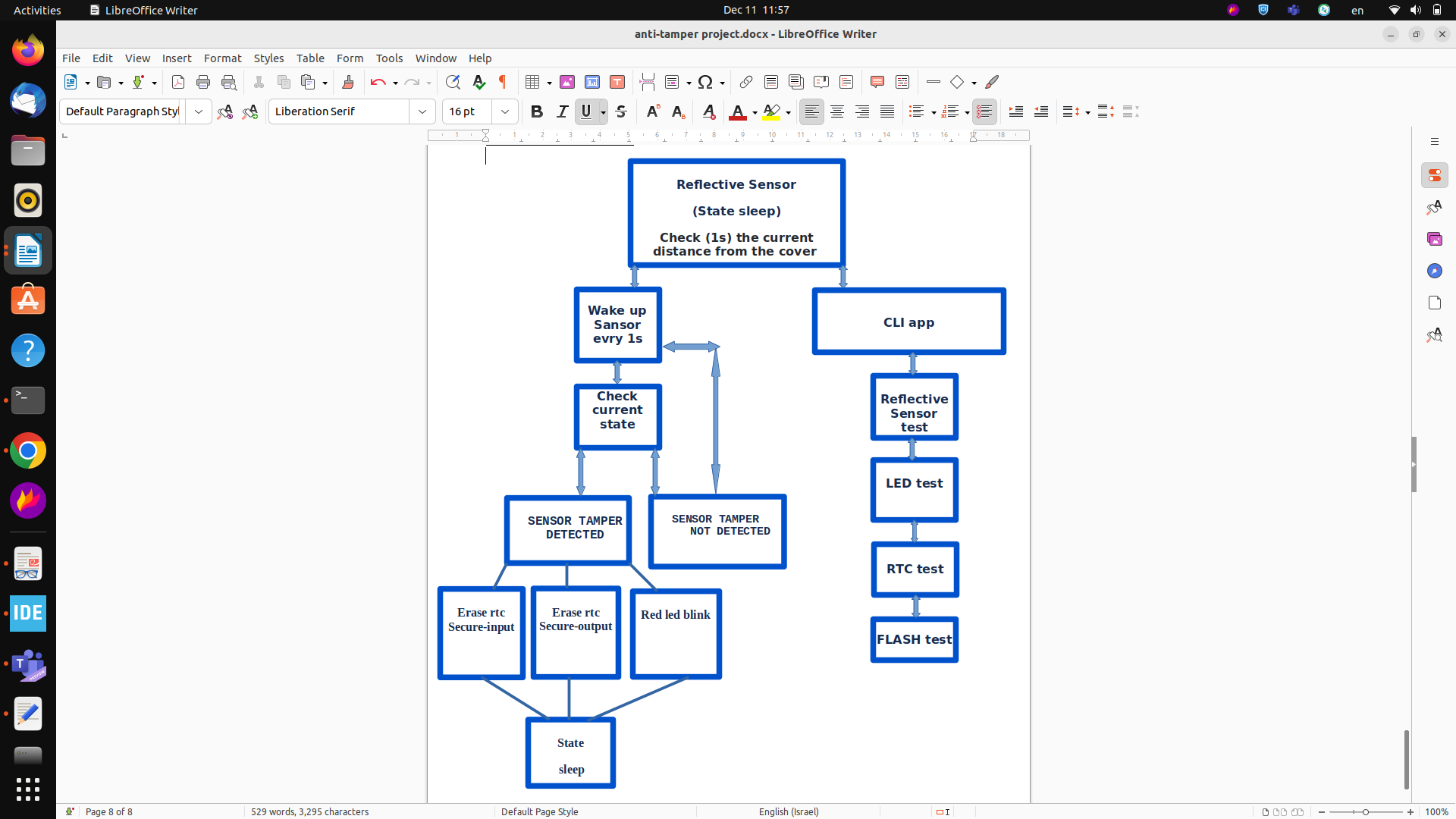
**Commuincation:**

* UART protocal
* GPIO protocal
* IC2 protocal
* SPI protocal

**Electrical diagram:**

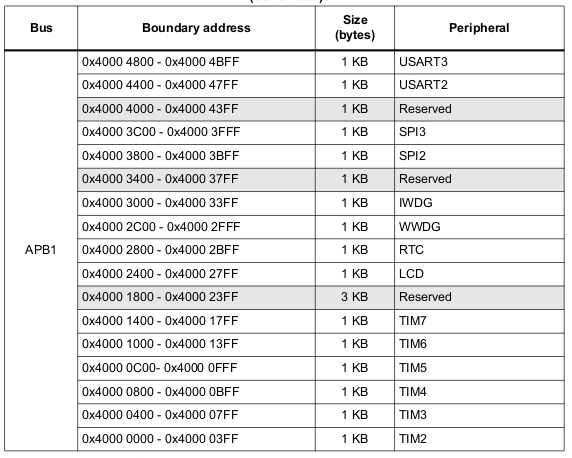


**BLOCK DIAGRAM:**



**Use cases**

**STM32L486xx memory map and peripheral register boundary addresses:**



**Anti-Tamper Sensor:**

**link:** [**https://www.onsemi.com/pdf/datasheet/qre1113-d.pdf**](https://www.onsemi.com/pdf/datasheet/qre1113-d.pdf)

* The regular state for Reflective Sensor – sleep!
* The Reflective Sensor should constantly check (1s) the current distance from the cover

**The system should have two events:**

* system stableed
* tamper event

**Reflective Sensor states:**

* SENSOR\_INITIALIZED
* SENSOR\_TAMPER\_DETECTED
* SENSOR\_TAMPER\_NOT\_DETECTED
* SENSOR\_ERROR

**if**

The Reflective Sensor dosen’t fined an tamper event it back to SENSOR\_INITIALIZED state

**else** **if**

The Reflective Sensor fined an tamper event:

**Event on Reflective Sensor/Switch Sensor top side:**

SENSOR\_TAMPER\_DETECTED!

1. Red Led switch blink
2. Erase the encryption keys - **Rtc Secure-input**
3. Erase the encryption keys - **Rtc Secure-output**
4. **optenal for other location?**

**Event on Reflective Switch Sensor buttom side:**

SENSOR\_TAMPER\_DETECTED!

1. Red Led switch blink
2. Erase encryption keys - **Rtc Secure-input**
3. Erase encryption keys - **Rtc Secure-output**
4. **optenal for other location**

**The system will try connect to the RTC (secureinput / secureoutput):**

* **the system try to connection for thr RTC in the power on:**

if it’s not connected - state SENSOR\_ERROR

else - SENSOR\_TAMPER\_NOT\_DETECTED

**Red Led:**

**Led states:**

* LED**\_**INITIALIZED\_READY
* LED\_EVENT\_BATTERY
* LED\_ERROR

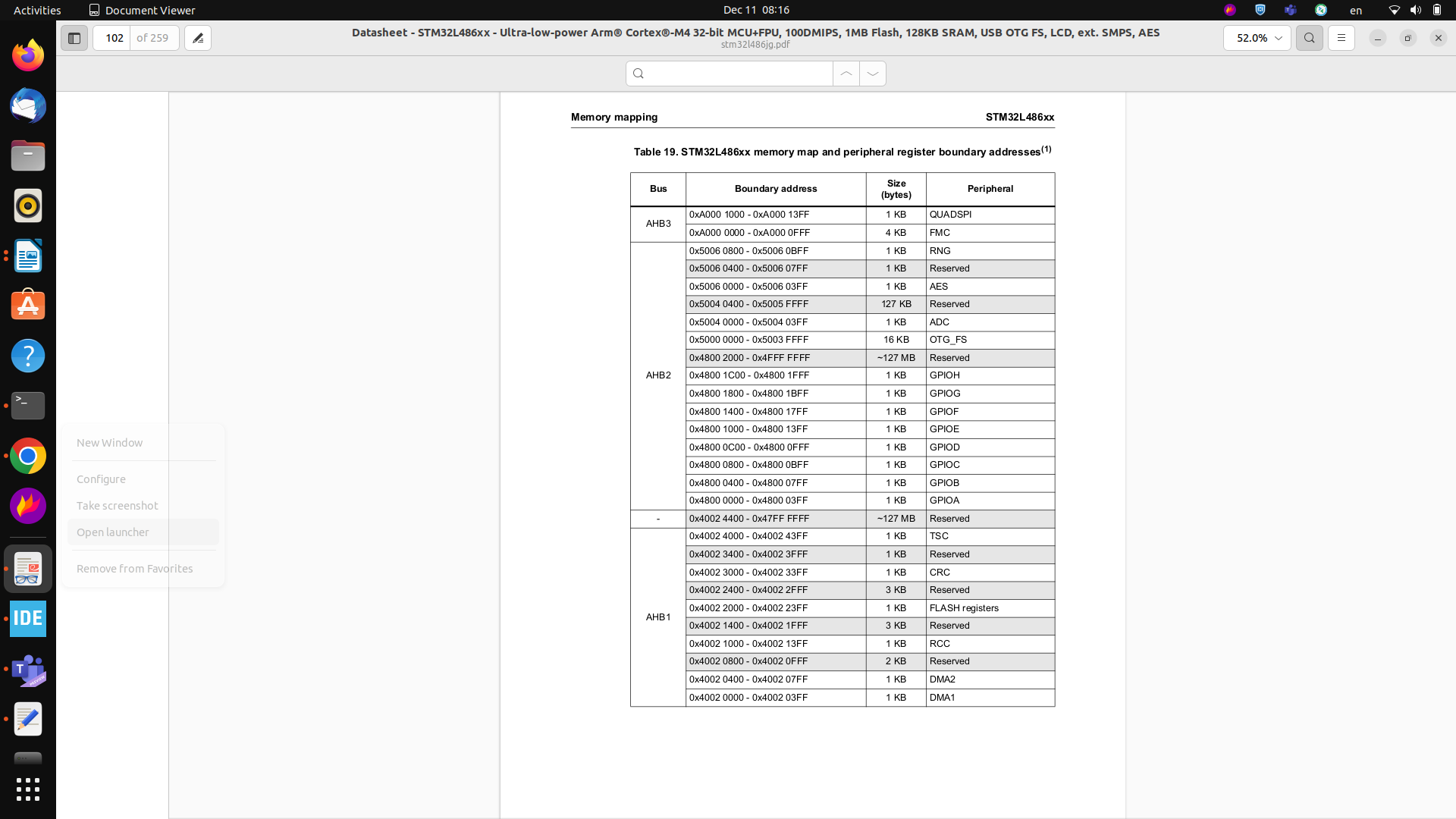
**state for red led on (got power from main battery and coin) –** LED**\_**INITIALIZED\_READY

**state for red led off (main and cion battery off) –** LED\_EVENT\_BATTERY

SENSOR\_TAMPER\_DETECTED!

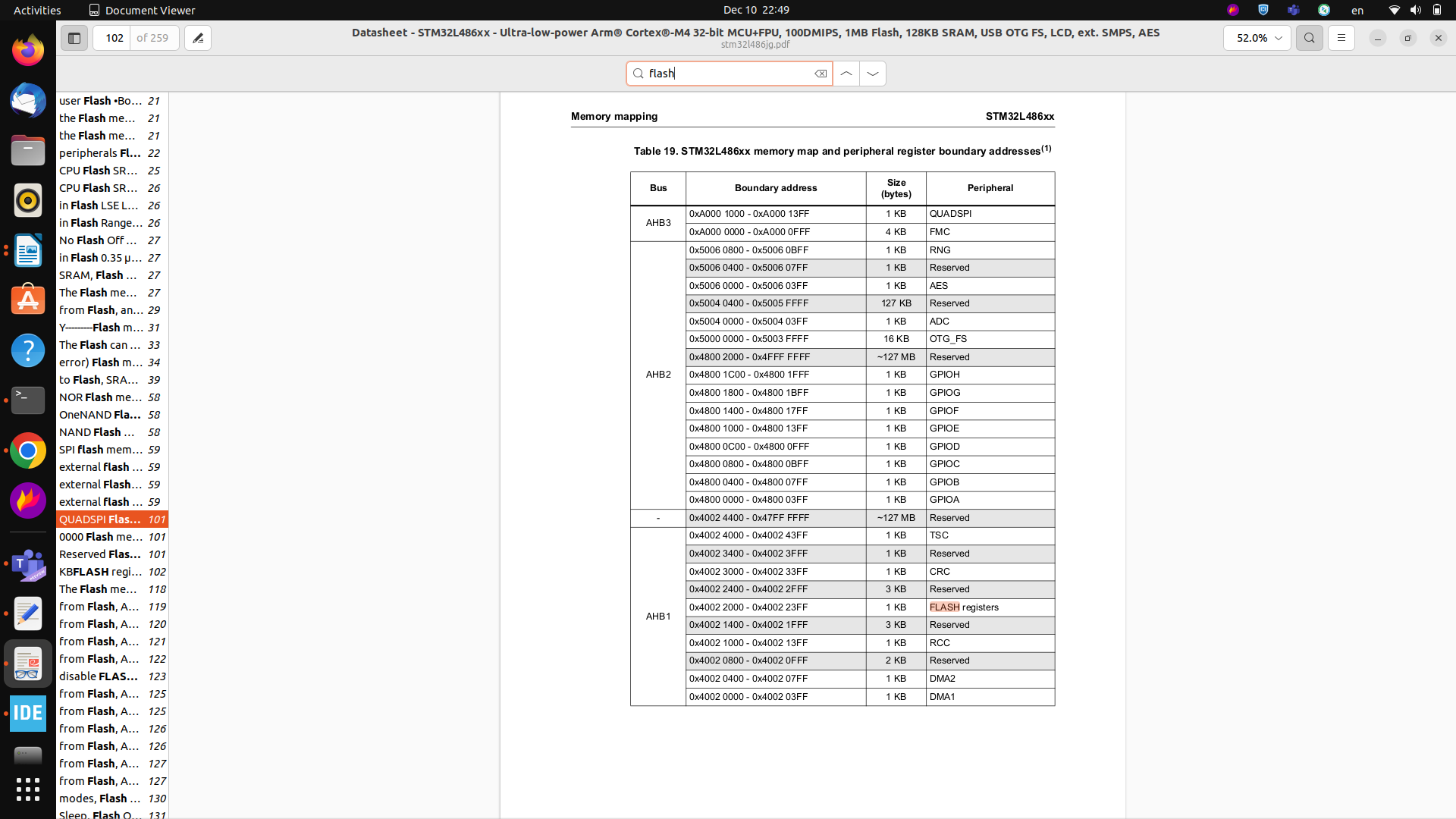
* 1. Erase the encryption keys - **Rtc Secure-input**
  2. Erase the encryption keys - **Rtc Secure-output**

**optenal for other location?**



**LOG:**

* app log print - prints all logs
* app save log - save logs in flash
* app log clear - clears log



**PWM:**

**SW1 / SW2 event**

HAL\_GPIO\_EXTI\_Callback(uint16\_tGPIO\_Pin)

SW1 – GPIO PIN 2

SW2 – GPIO PIN 8

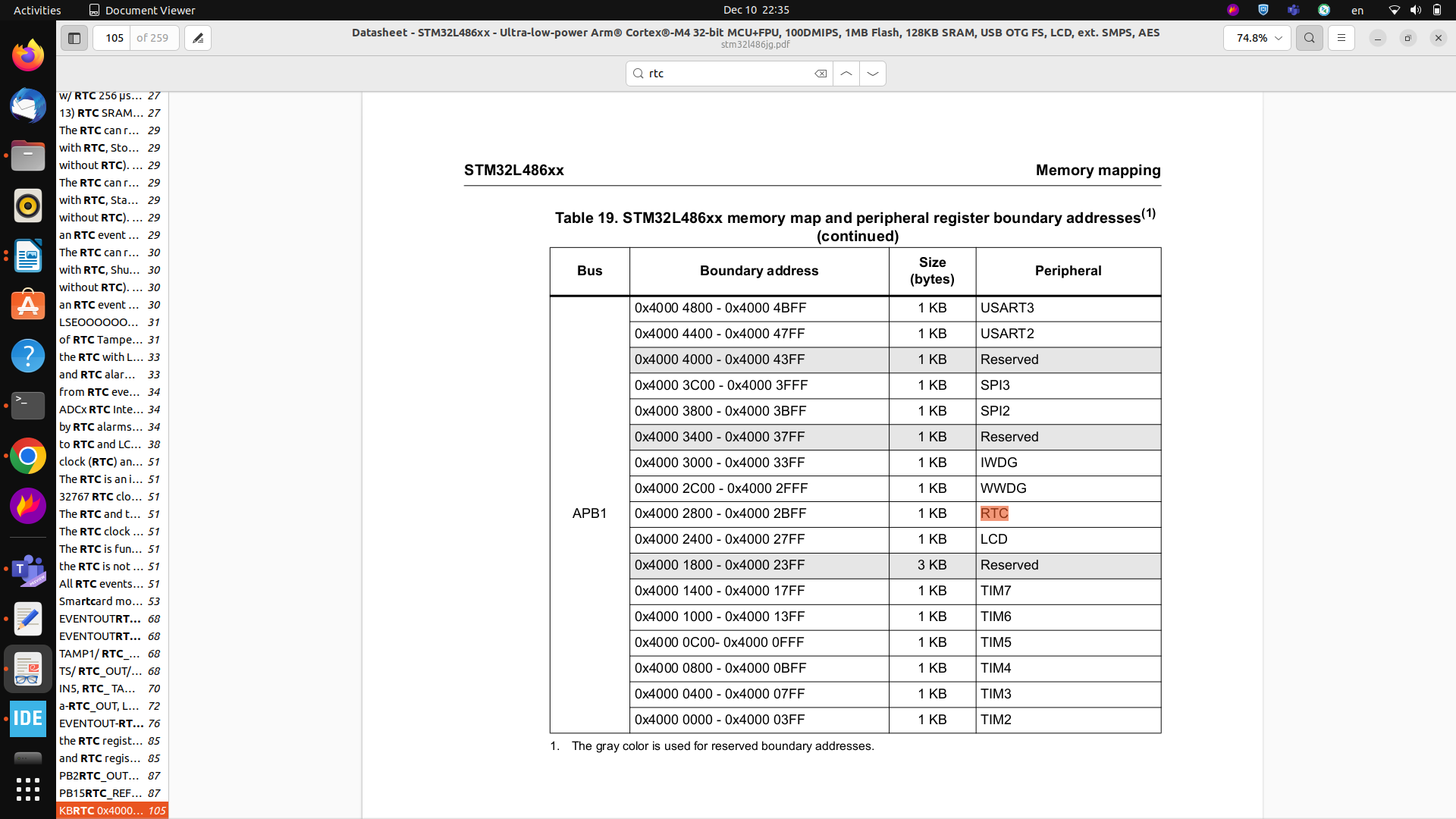
**UART Console:**

UART3 – PB10/PB11

app fpr Cli Container

Open UART console from Linux PC to the device:

Picocom -b 115200 /dev/ttyUSB0



**Reflective Sensor test**

app set distance (Threshold)

app get distance (TAMPER\_DETECTED / TAMPER\_NOT\_DETECTED)

**LED test**

gpio conf GPIOC PIN 7 out

gpio set GPIOC PIN 7 < value 0 or 1> # in this case, LED is inversed logic, turn "on" on 0

gpio conf GPIOC PIN 7 out gpio set GPIOC PIN 7 0 gpio set GPIOC PIN 7 1

**RTC test**

app rtc set time

app rtc get time

**FLASH test:**

app write to flash

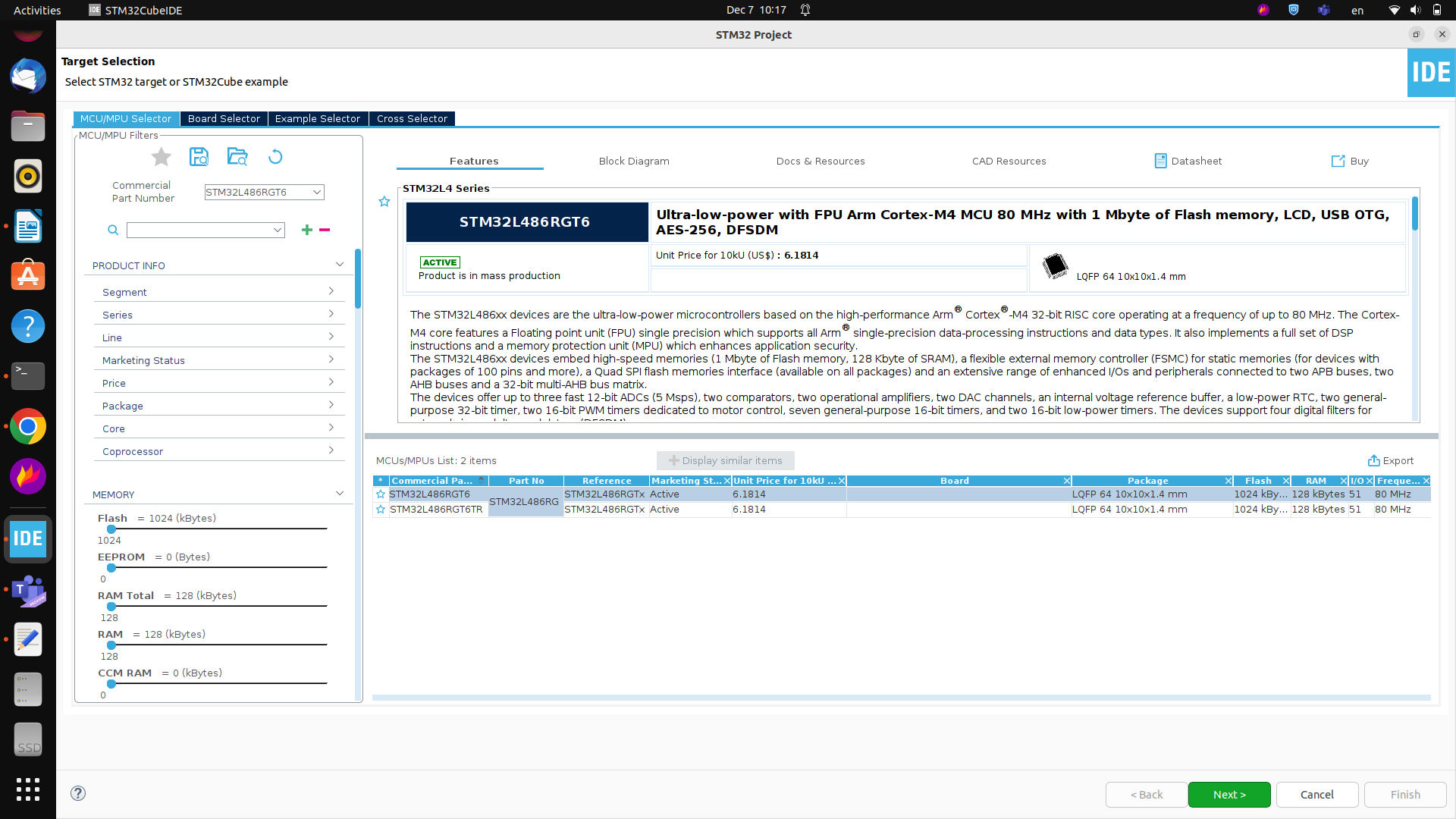
app read flash data

app clear flash data

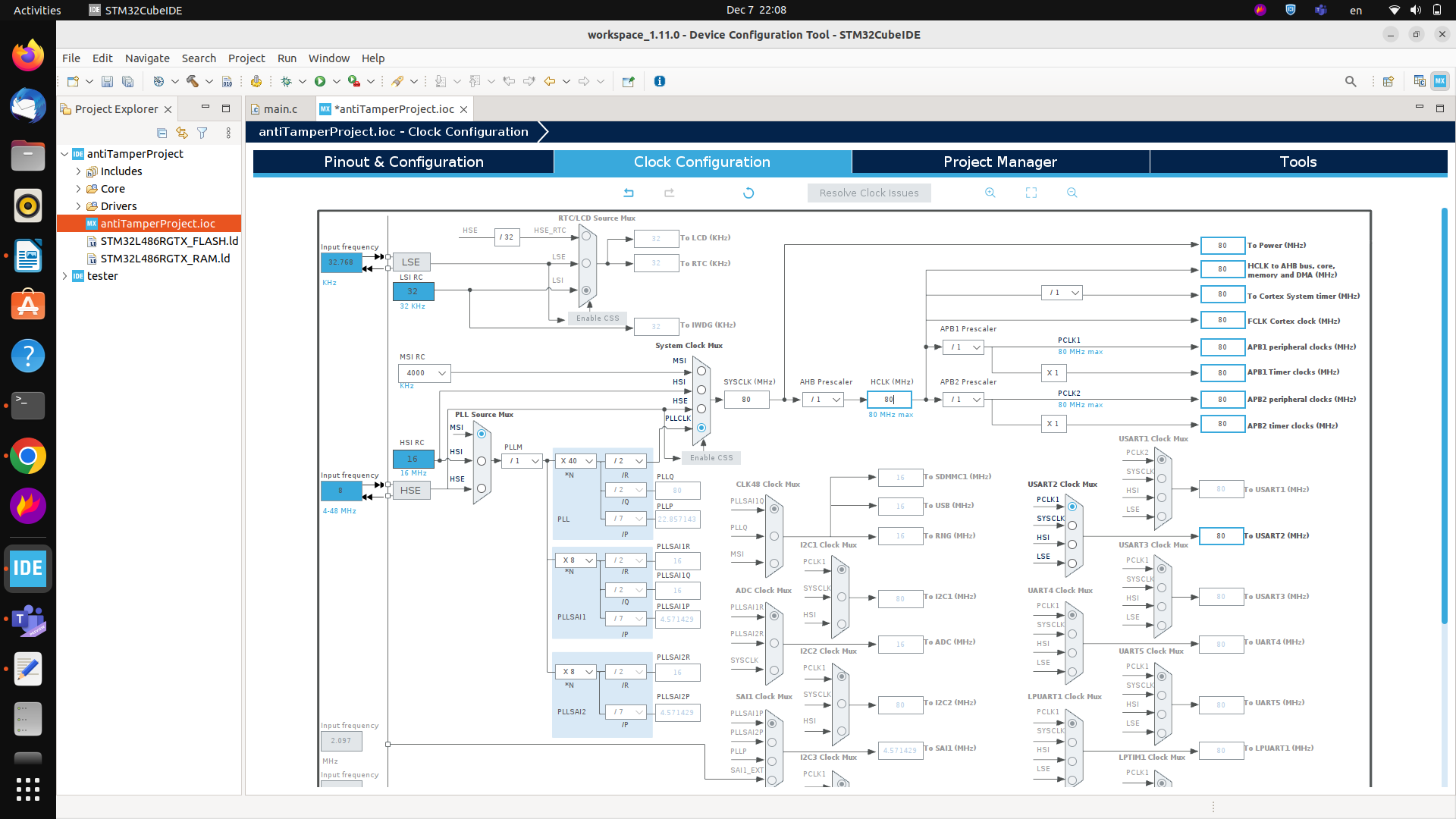
**How to use cube in order to set clock rates:**

**open new project:**



****

**Set internal clock 80 Mhz max :**

****