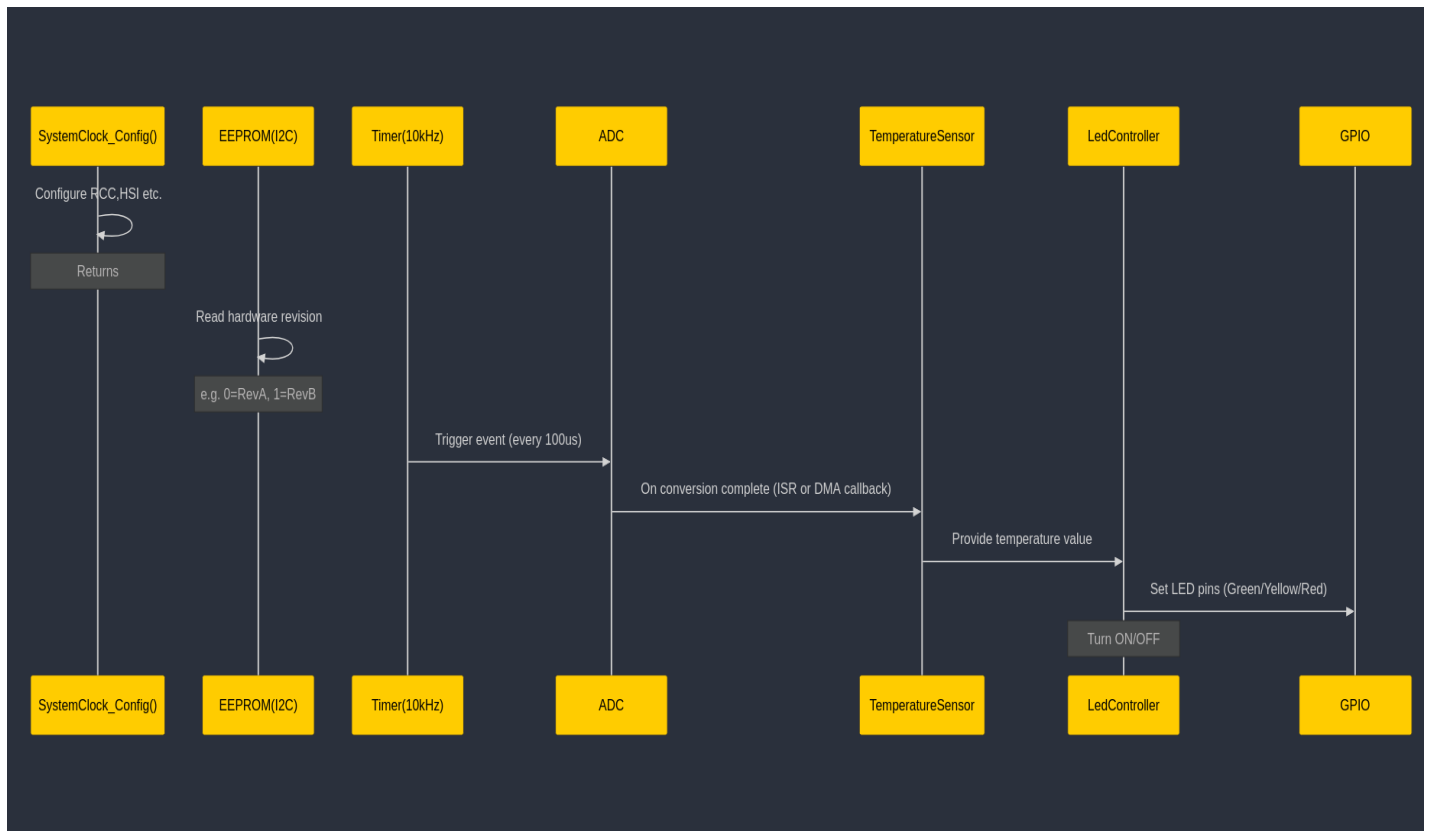The provided diagrams represent the **architecture of the system**, which is based on an **STM32 microcontroller**.

- A **SysML-style block diagram** (using flowchart syntax) illustrates the high-level **system context**.
- A **UML class diagram** (using classDiagram syntax) models the **C++ code structure**.

To **save time**, some descriptions are provided **as comments directly in the code** rather than in the diagrams.

SysML-like system overview:

UML class diagram

Main

+ main() : : int

---

uses

Eeprom

+ Eeprom_I2C_Init() : : void
+ Eeprom_ReadBytes(addr: size_t, buf: uint8_t*, len: size_t) : : bool

---

uses

«abstract»
TemperatureSensor

- TemperatureReadyCallback callback

+ init() : : void
+ startSampling() : : void
+ setCallback(cb: TemperatureReadyCallback) : : void

---

uses

LedController

+ init() : : void
+ setGreenOn(on: bool) : : void
+ setYellowOn(on: bool) : : void
+ setRedOn(on: bool) : : void
+ update(temp: float) : : void

RevASensor

- internalData : ...

+ init() : : void
+ startSampling() : : void
+ setCallback(cb) : : void

RevBSensor

- internalData : ...

+ init() : : void
+ startSampling() : : void
+ setCallback(cb) : : void

---

uses

Gpio

+ Gpio_InitPin(pin: GpioPin) : : void
+ Gpio_WritePin(pin: GpioPin, state: bool) : : void

---

uses

Adc

+ Adc_Init(cb: AdcCallback) : : void
+ Adc_Start() : : void
+ Adc_SimulateConversion(val: uint16_t) : : void

---

triggered by

Timer

+ Timer_Init100us(cb: TimerCallback) : : void
+ Timer_SimulateTick() : : void