RELÓGIO ALARME COM STM32F103C6

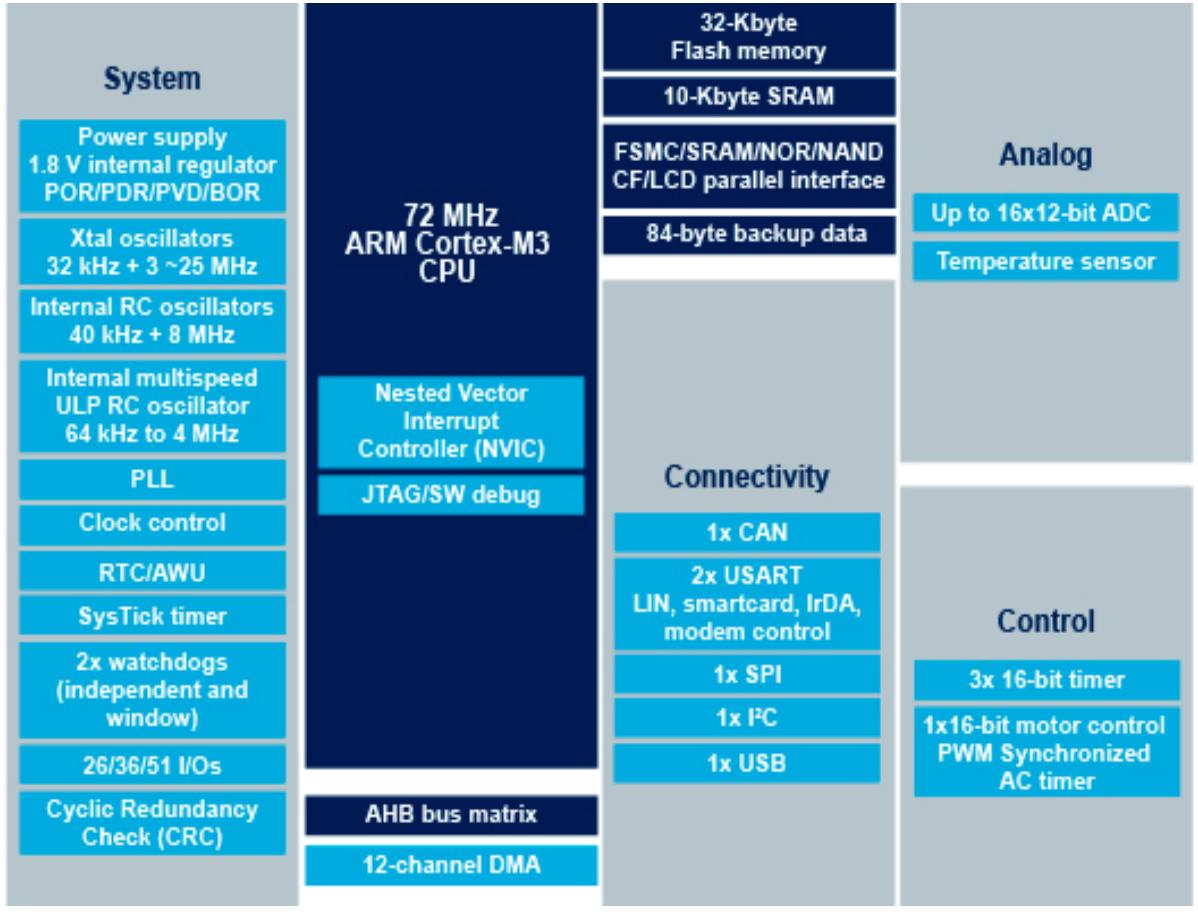
Equipe:

1. John Vasconcelos dos Santos — Matrícula: 414953
2. Renato Avelino — Matrícula: 485369
3. Rayane Gadelha Melo de Lima — Matrícula: 368610

Disciplina: Sistemas Microprocessados.
Data de Entrega do Projeto: 11 de abril de 2021

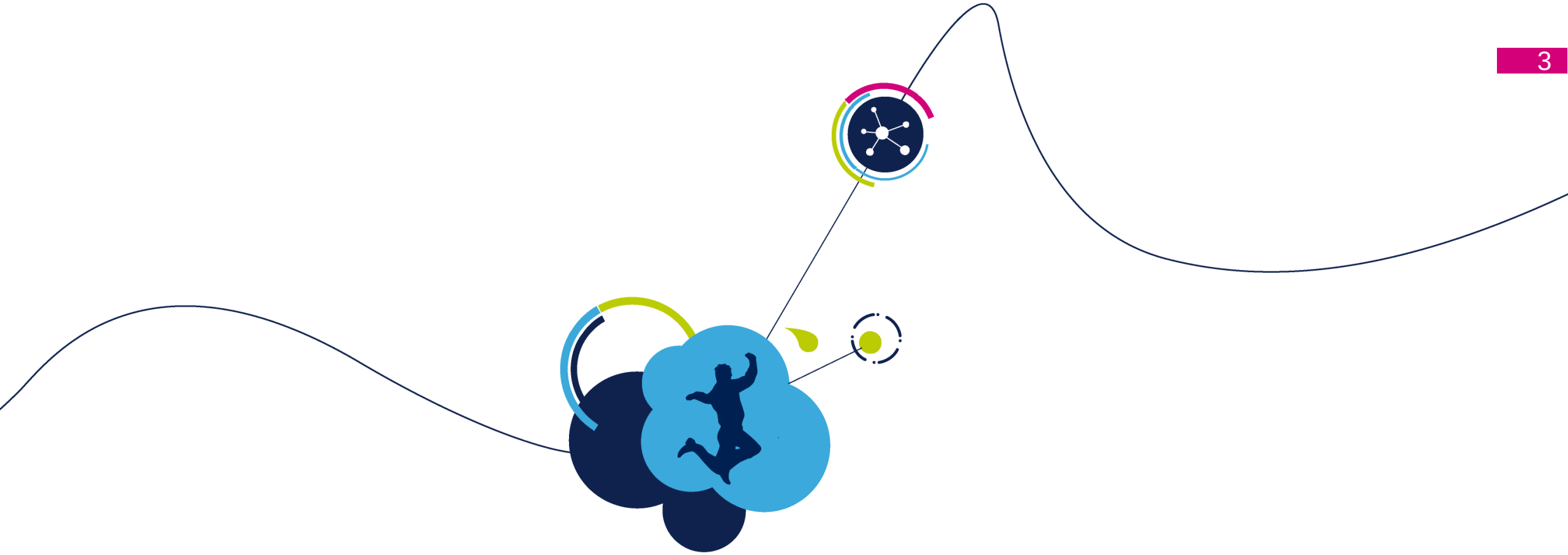
Fortaleza 2021

Microprocessador STM32F103XX



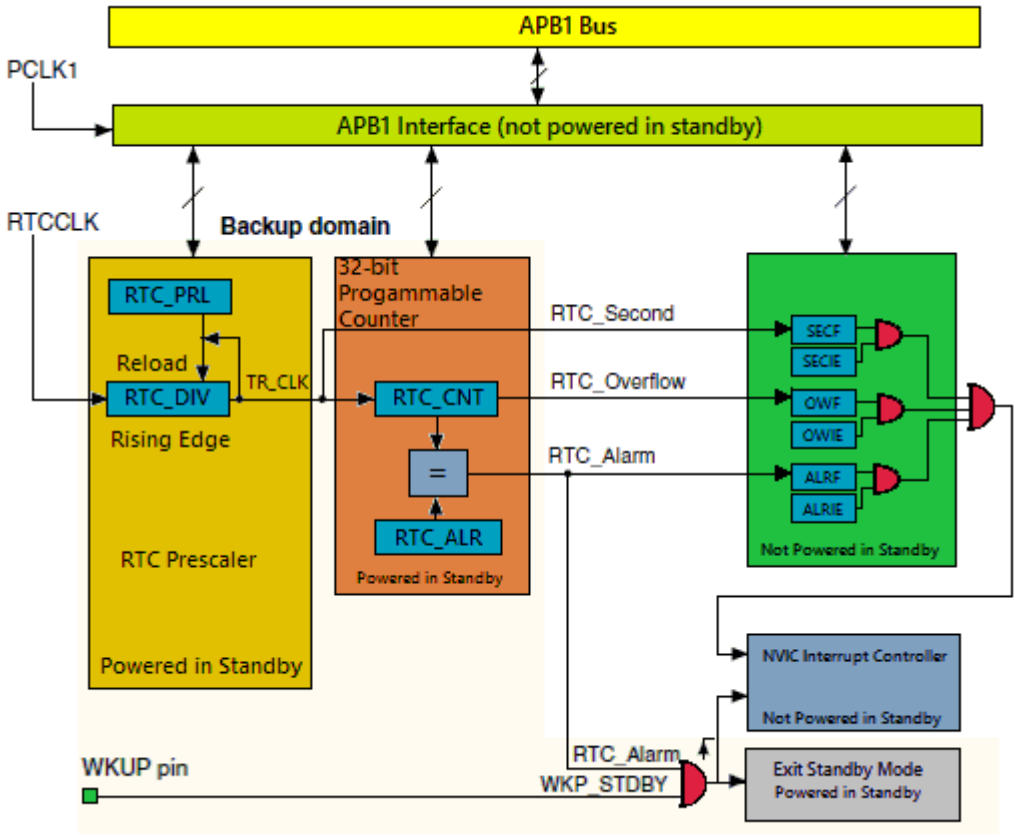
- RISC de 32 bits ARM® Cortex ™ - M3;
- Frequência de 72 MHz;
- Memória Flash incorporada de até 32 Kbytes e SRAM de até 6 Kbytes);
- 02 ADCs de 12 bits;
- 03 temporizadores de 16 bits de uso geral;
- 01 temporizador PWM;
- 02 I2Cs e SPIs, 03 USARTs, um USB e um CAN

life.augmented



Real Time Clock (RTC):

Microprocessador STM32F103C6



- Prescaler programável
- Contador programável de 32 bits para medição de longo prazo
- Dois relógios separados: PCLK1 para a interface APB1 e relógio RTC.
- A fonte de relógio RTC pode ser qualquer uma das seguintes: HSE clock divided by 128, LSE oscillator clock E LSI oscillator clock.
- Três linhas de interrupção dedicadas: Interrupção de alarme, para gerar uma interrupção de alarme programável por software, Interrupção de segundos, para gerar um sinal de interrupção periódica com uma duração de período programável (até 1 segundo), Interrupção de estouro, para detectar quando o contador programável interno chega a zero.

OBJETIVOS

Projeta e simular no Proteus de um relógio digital com alarme utilizado o microprocessador STM32F103C6.

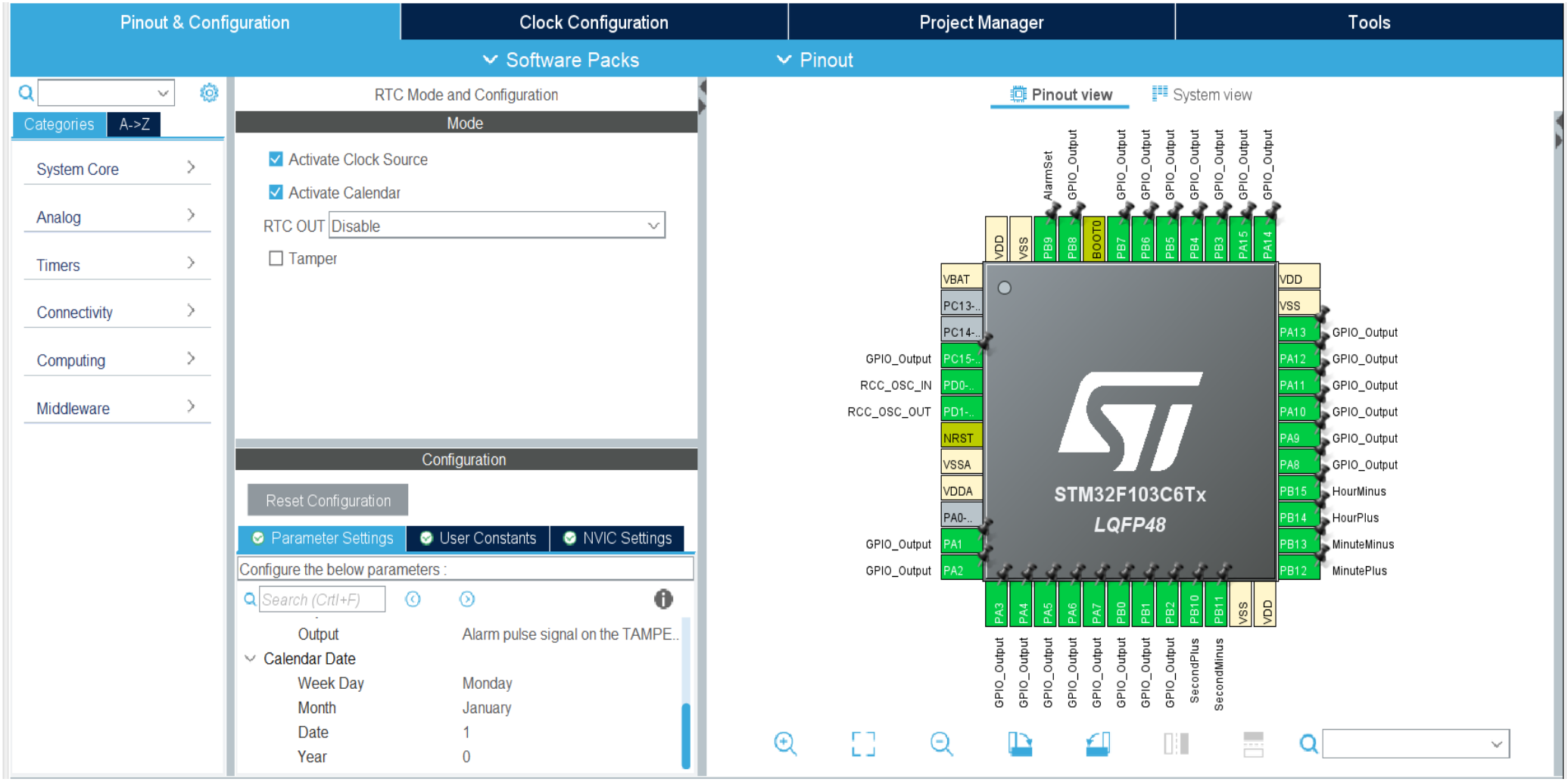
1. MATERIAIS

- Simulador de circuitos digitais Proteus;
- Microprocessador STM32F103C6;
- Chave digital micro Switch;
- 07 chaves Botões circuitos integrais e digitais;
- 06 Display 7 segmentos;
- 01 Led vermelho;
- 07 Resistores de 10kΩ;
- 08 Resistores de 330kΩ.

Configuração das portas do microprocessador STM32F103C6

Porta	*Configuração*	*Utilização*
PA01-PA15	GPIO_OUTPUT	Representação das horas, minutos e segundos no Display 7 segmentos.
PB0-PB08	GPIO_OUTPUT	Representação das horas, minutos e segundos no Display 7 segmentos.
PB009-PB015	GPIO_EXTI	Interrupção da função principal através de chaves digitais.
PC15	GPIO_OUTPUT	Acendimento de um LED quando o tempo do relógio for igual do Alarme.
PD0-PD01	RCC_OSC	Cristal ressonador interno

Configuração RTC



Código

/* **Declaração de variáveis globais do RTC** */

```
/* Private variables ---------------------------------------------------------*/
RTC_HandleTypeDef hrtc;

/* USER CODE BEGIN PV */

uint8_t alarmflag = 0;
RTC_TimeTypeDef sTime = {0};
RTC_AlarmTypeDef sAlarm = {0};

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
```

Código


```
/* SET do valor inicial do alarme */

  /* Infinite loop */

  sAlarm.AlarmTime.Hours = 9;
  sAlarm.AlarmTime.Minutes = 16;
  sAlarm.AlarmTime.Seconds = 20;
      sAlarm.Alarm =
RTC_ALARM_A;

  /* USER CODE BEGIN WHILE */
```

```
/** Initialize RTC and set the Time and Date  */
  sTime.Hours = 0x9;
  sTime.Minutes = 0x16;
  sTime.Seconds = 0x20;

  if (HAL_RTC_SetTime(&hrtc, &sTime,
RTC_FORMAT_BCD) != HAL_OK)
  {
    Error_Handler();
  }
  DateToUpdate.WeekDay =
RTC_WEEKDAY_MONDAY;
  DateToUpdate.Month =
RTC_MONTH_JANUARY;
  DateToUpdate.Date = 0x1;
      DateToUpdate.Year = 0x0;
```

Código

/* Leitura do valor contido no Time do RTC e no Alarme e transformação de binária decimal (BCD) em binário para o Display 7 segmentos*/

```
while (1)
 {
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
        HAL_Delay(100);
        RTC_TimeTypeDef tmpTime;
        RTC_AlarmTypeDef tmpAlarm;
        HAL_RTC_GetTime(&hrtc, &tmpTime, RTC_FORMAT_BIN);
        HAL_RTC_GetAlarm(&hrtc, &tmpAlarm, RTC_ALARM_A, RTC_FORMAT_BIN);

        if(alarmflag == 0) {
                uint8_t secFirstDigit = tmpTime.Seconds%10;
                uint8_t secSecondDigit = tmpTime.Seconds/10;

                //uint8_t secFirstDigit = alarmflag?(tmpAlarm.AlarmTime.Seconds%10):(tmpTime.Seconds%10);
                //uint8_t secSecondDigit = alarmflag?(tmpAlarm.AlarmTime.Seconds/10):(tmpTime.Seconds/10);
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, secFirstDigit & 0x01);
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, secFirstDigit & 0x02);
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, secFirstDigit & 0x04);
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, secFirstDigit & 0x08);

                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, secSecondDigit & 0x01);
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, secSecondDigit & 0x02);
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, secSecondDigit & 0x04);
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, secSecondDigit & 0x08);
```

Código

/* Função comparativa entre Time e alarme, se for igual faz piscar um Led na Porta C Pino 15*/

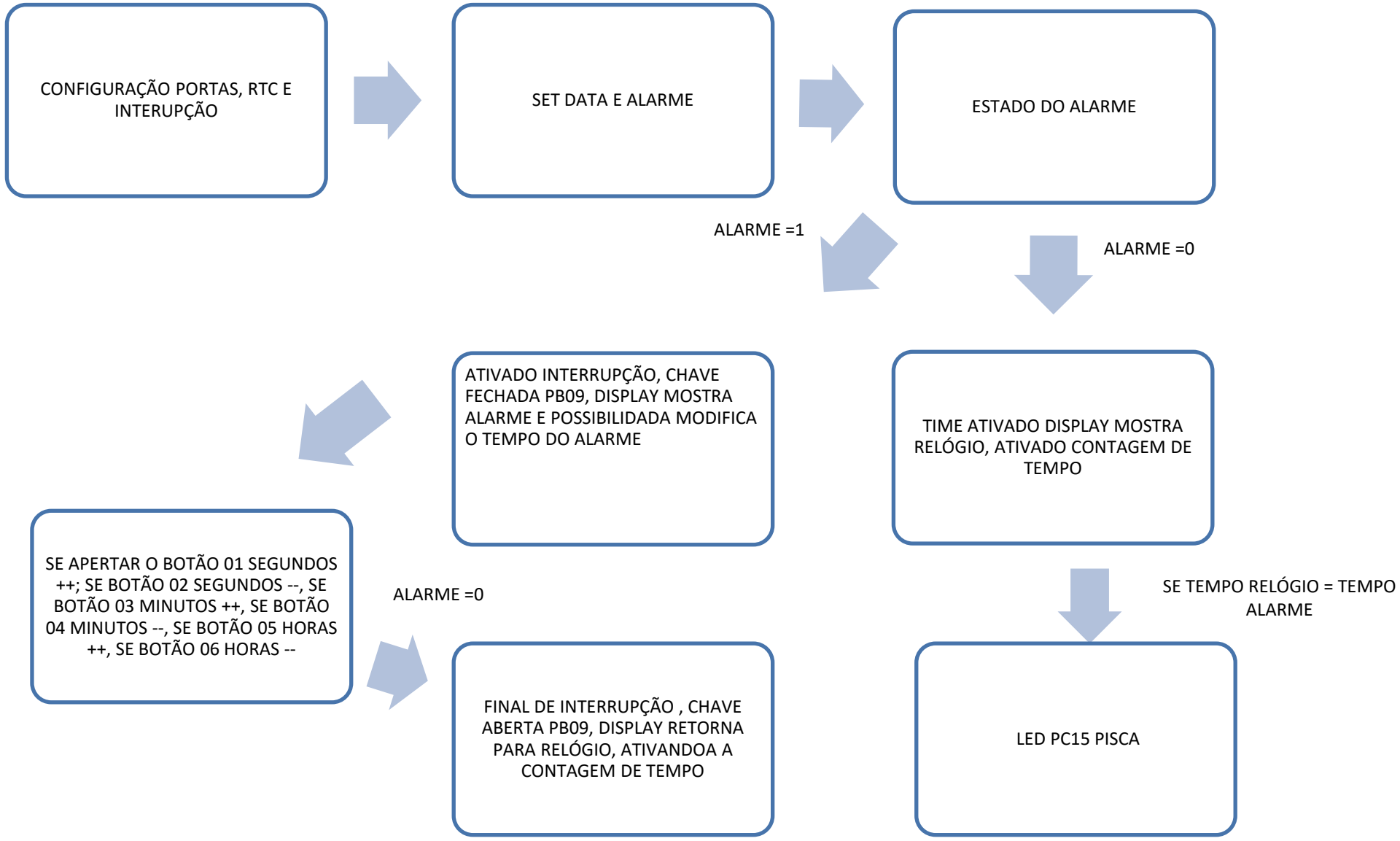
```
if (tmpAlarm.AlarmTime.Hours == tmpTime.Hours) {
                if (tmpAlarm.AlarmTime.Minutes == tmpTime.Minutes) {

if(tmpAlarm.AlarmTime.Seconds == tmpTime.Seconds) {

HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_15);

HAL_Delay(200);
```

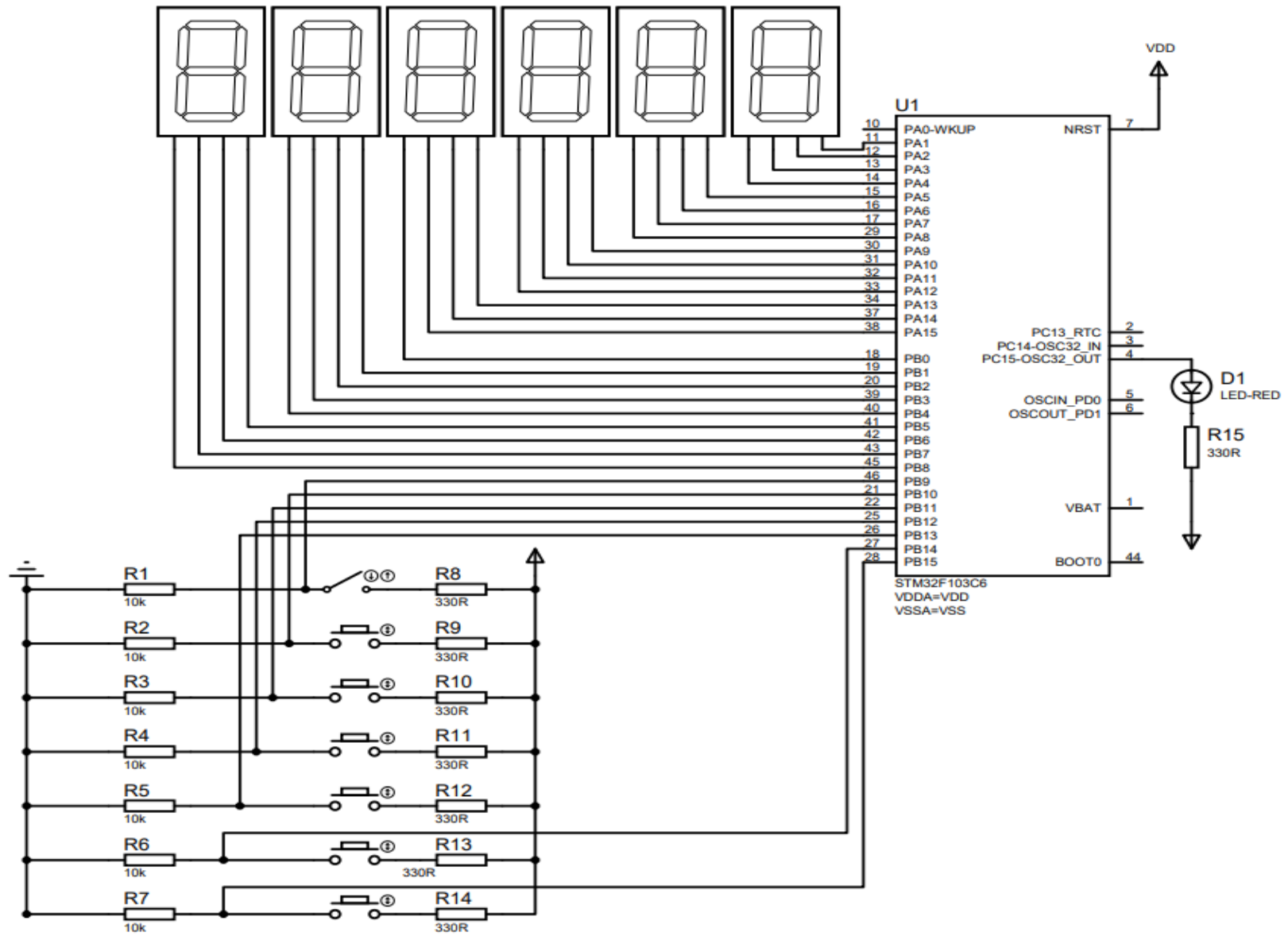
/ Interrupção do Time do RTC através da chave no Porta B Pino 09, adicionando ou subtraindo valores conforme o botão acionado */**

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {

        UNUSED(GPIO_Pin);
        switch(GPIO_Pin) {
                case AlarmSet_Pin: if (alarmflag) alarmflag = 0; else alarmflag = 1; break;
                case SecondPlus_Pin: if (alarmflag) sAlarm.AlarmTime.Seconds++; break;
                case MinutePlus_Pin: if (alarmflag) sAlarm.AlarmTime.Minutes++; break;
                case HourPlus_Pin: if (alarmflag) sAlarm.AlarmTime.Hours++; break;
                case SecondMinus_Pin: if (alarmflag) sAlarm.AlarmTime.Seconds--; break;
                case MinuteMinus_Pin: if (alarmflag) sAlarm.AlarmTime.Minutes--; break;
                case HourMinus_Pin: if (alarmflag) sAlarm.AlarmTime.Hours--; break;
        }
        if(alarmflag) HAL_RTC_SetAlarm(&hrtc, &sAlarm, RTC_FORMAT_BIN);

}
```

Dispositivo	*Configuração*	*Utilização*
Display 01	Interligado nas PA01-PA04	segundos
Display 02	Interligado nas PA05-PA08	segundos
Display 03	Interligado nas PA09-PA12	Minutos
Display 04	Interligado nas PA13-PB0	Minutos
Display 05	Interligado nas PB01-PA04	Horas
Botão 01	*Interligado nas PB10*	*set segundos alarme para mais*
Botão 02	Interligado nas PB11	set segundos alarme para menos
Dispositivo	Configuração	Utilização
Botão 03	Interligado nas PB12	set minutos alarme para mais
Botão 04	Interligado nas PB13	set minutos alarme para menos
Botão 05	Interligado nas PB14	set horas alarme para mais
Botão 05	Interligado nas PB15	set horas alarme para menos
Led	Interligado nas PC15	Pisca quando alarme = Time

Diagrama




U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
STM32F103C6
VDDA=VDD
VSSA=VSS
VDD
D1
LED-RED
R15
330R
R1
10k
R2
10k
R3
10k
R4
10k
R5
10k
R6
10k
R7
10k
R8
330R
R9
330R
R10
330R
R11
330R
R12
330R
R13
330R
R14
330R

Thank you

life.augmented