Lab 3: Radio-controlled Clock with DCF77

Dokumentation

Tim Jauch

Ergün Bickici

Contents

List of Figures

List of Tables

1 Requirements

- The first line of the Dragon12's LCD shall display the time in the format: `hh:mm:ss`. The second line shall show the date in the format: `dd.mm.yyyy`. Manual time setting, as implemented in Lab 2, is no longer required.
- Since DCF77 signal reception may occasionally fail (e.g., due to poor radio conditions in reinforced concrete buildings or during transmitter maintenance), the internal timer-based clock from Lab 2 remains active. The DCF77 signal is only used to adjust the time once per minute and to display the date. If the DCF77 data becomes unavailable, the system continues running solely on the internal timer.
- The internal timer clock shall toggle the LED connected to Port B.0 once per second.
- The LED on Port B.1 shall reflect the raw DCF77 signal: it turns on when the signal is low, and off when it is high. As a result, this LED will toggle once per second when the DCF77 signal is being received.
- The LED on Port B.3 shall be turned on when a complete and valid DCF77 time and date message has been successfully decoded. It turns off immediately if no valid data is received or the data is corrupted.
- The DCF77 signal is connected to Port H.0 on the real Dragon12 board. Due to potential signal spikes and jittering edges, it is recommended to poll the signal every 10 ms instead of relying on Port H interrupts.

2 User Interface Description

2.1 LCD Display

- The first line of the Dragon12 LCD displays the current time in the format `hh:mm:ss`.
- The second line shows the date in the format `dd.mm.yyyy`.
- The day of the week (e.g., `Sun`, `Mon`, ...) is displayed in front of the date.
- The active time zone is indicated by either `DE` or `US` on the LCD.

2.2 LED Indicators

- The LED connected to Port `B.0` toggles once per second, driven by the internal timer clock.
- The LED on Port `B.1` reflects the raw DCF77 signal level: it turns on when the signal is low and off when the signal is high. This results in a 1 Hz blinking pattern during proper DCF77 reception.
- The LED on Port `B.2` lights up when an error is detected and remains on until valid data is received.
- The LED on Port `B.3` turns on when a complete and valid DCF77 time and date frame is successfully decoded. It turns off immediately if the received data is invalid or missing.

2.3 Control Button

- The button connected to `PTH3` toggles between the MET and EST time zones.

3 Debugging

No debugging was required, as all provided code segments functioned correctly. Only the necessary extensions to the `dcf77.c` and `clock.c` files had to be implemented.

4 Data Dictionary

4.1 List of Global Variables

Module(s)	Variable	Type	Purpose
`clock.c`, `main.c`	`clockEvent`	`enum`	Defines clock event types (e.g., `SET`, `TICK`, `ALARM`)
`clock.c`, `dcf77.c`	`dcf77TimeZone`	`enum`	Indicates selected time zone: `MET` or `EST`
`clock.c`, `dcf77.c`, `main.c`	`dcf77Event`	`enum`	Represents different signal states or transitions from the DCF77 decoder

Table 1: Global variables used across modules

4.2 Hardware Resources

Module	HCS12 Hardware	Purpose
`dcf77.c`	Port H	Receives the DCF77 antenna signal via input pin
`led.asm`	Port B	Controls status LEDs (DCF77 signal, timer pulses, error states)
`lcd.asm`	Dragon12 LCD	Displays time, date, weekday, and time zone

Table 2: Hardware resources accessed by software modules

5 Module overview

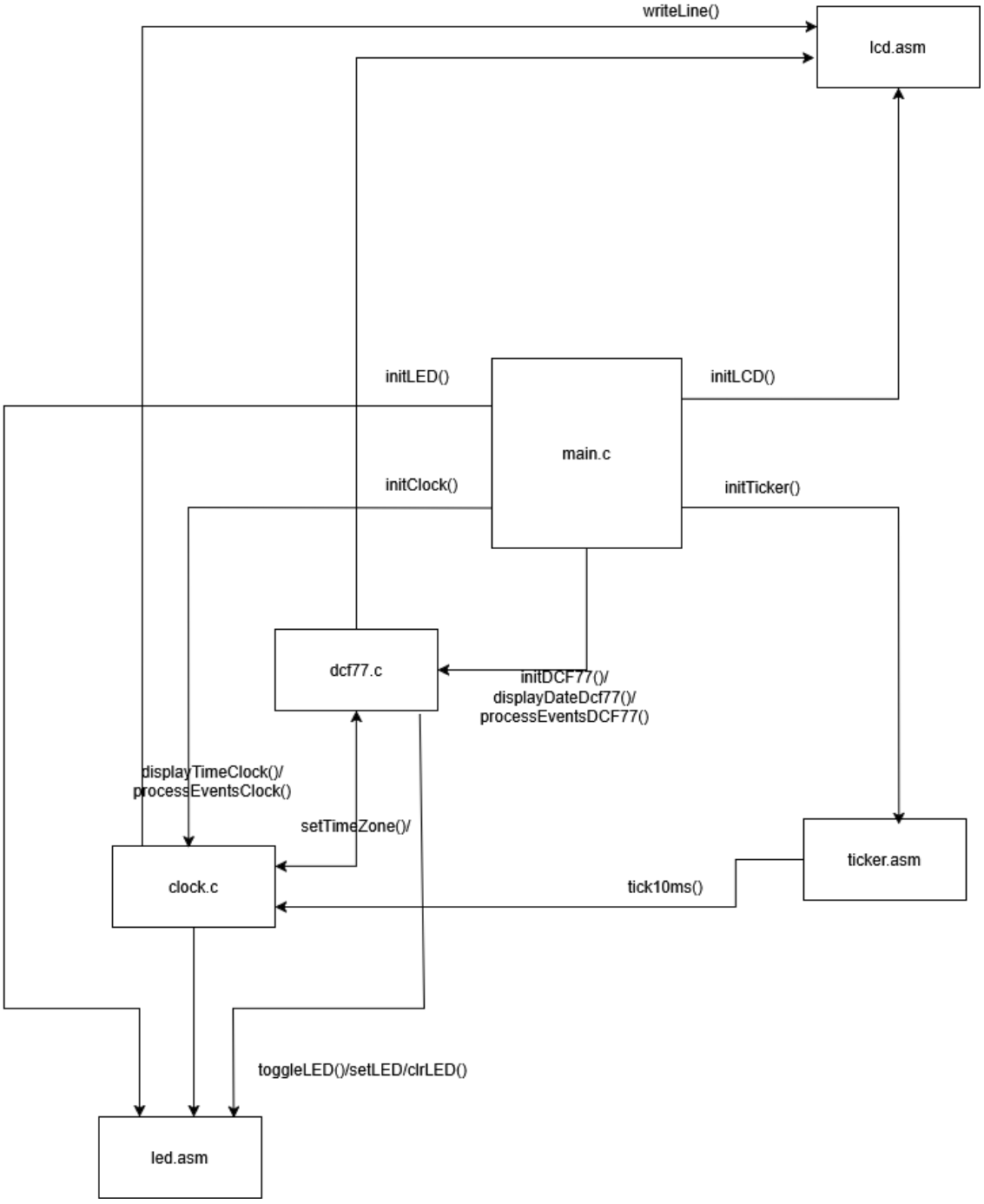

Figure 1: Module Overview

6 Interface Descriptions of All Subroutines

6.1 main.c

- **main()** Initializes all modules and calls the corresponding subroutines.

  Parameters: –
  Returns: –

6.2 clock.c

- **initClock()** Loads the initial time and weekday.

  Parameters: –
  Returns: –

- **tick10ms()** Called every 10 ms to advance internal clock logic.

  Parameters: –
  Returns: –

- **processEventsClock(clockEvent)** Called every second to update internal time values based on the event.

  Parameters: `clockEvent` (typically `SECONDTICK`)
  Returns: –

- **setClock(hours, minutes, seconds)** Allows other modules (e.g., DCF77) to set the time.

  Parameters: `int hours, minutes, seconds`
  Returns: –

- **tzName()** Returns the current time zone name.

  Parameters: –
  Returns: `const char*`

- **tzHours()** Returns the current time zone offset in hours.

  Parameters: –
  Returns: `int`

- **displayTimeClock()** Displays time on LCD (line 0).

  Parameters: –
  Returns: –

- **time()** Returns the current CPU time base in milliseconds.

  Parameters: –
  Returns: `unsigned long`

6.3 dcf77.c

- **initializePort()** Initializes the input port for the DCF77 signal.

  Parameters: –

  Returns: –

- **readPort()** Reads the DCF77 signal level from the input port.

  Parameters: –

  Returns: `int` (0 = Low, > 0 = High)

- **initDCF77()** Initializes the DCF77 module.

  Parameters: –

  Returns: –

- **getDaysInMonth(month, year)** Returns number of days for a given month and year.

  Parameters: `int month, year`

  Returns: `int`

- **updateDate()** Updates date and time based on current time zone.

  Parameters: –

  Returns: –

- **displayDateDcf77()** Displays the date decoded from the DCF77 signal on LCD (line 1).

  Parameters: –

  Returns: –

- **sampleSignalDCF77(currentTime)** Evaluates DCF77 signal and detects events. Called every 10 ms.

  Parameters: `unsigned long currentTime`

  Returns: `dcf77Event`

- **processEventsDCF77(event)** Processes decoded events using DCF77 state machine.

  Parameters: `dcf77Event event`

  Returns: –

- **setDateDcf77()** Transfers data from buffer to global variables.

  Parameters: –

  Returns: `int` (1 = valid, 0 = invalid)

- **verifyValue(start, end)** Decodes and checks BCD-coded value including parity.

  Parameters: `unsigned char start, end`

  Returns: `int` (1 = valid, 0 = invalid)

- **setTimeZone(timeZone, hours, minutes)** Sets the system time zone.

  Parameters: `TIMEZONE timeZone, char hours, minutes`

  Returns: –

7 Flow Charts

7.1 Main

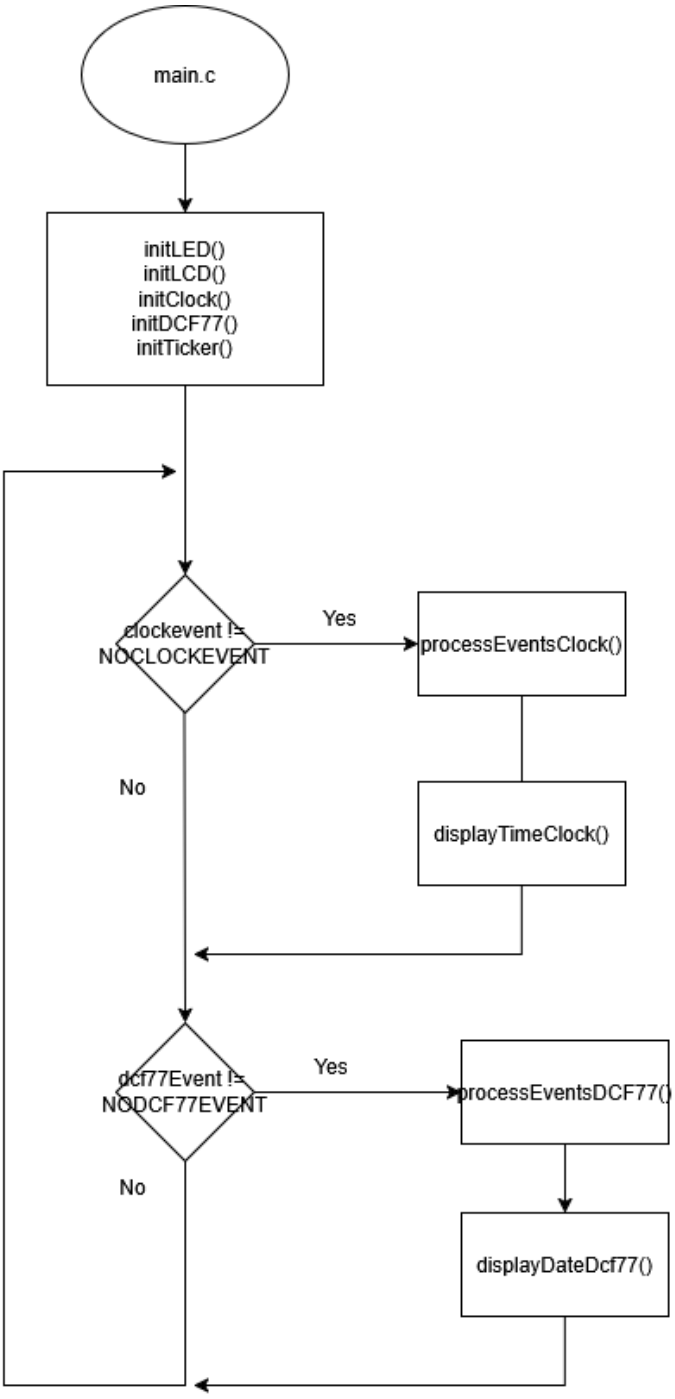

Figure 2: Main

7.2 Clock

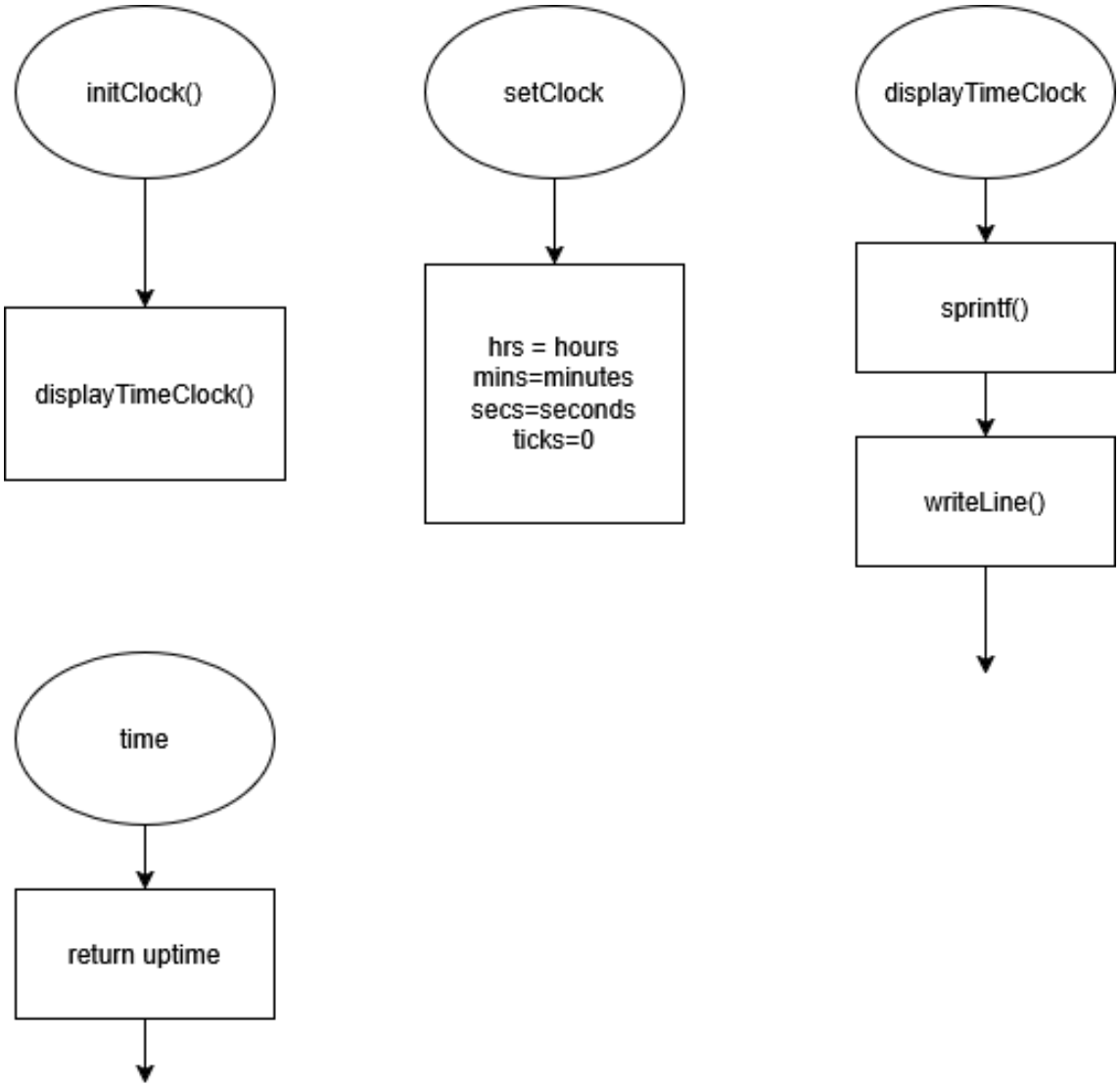

Figure 3: Clock 1

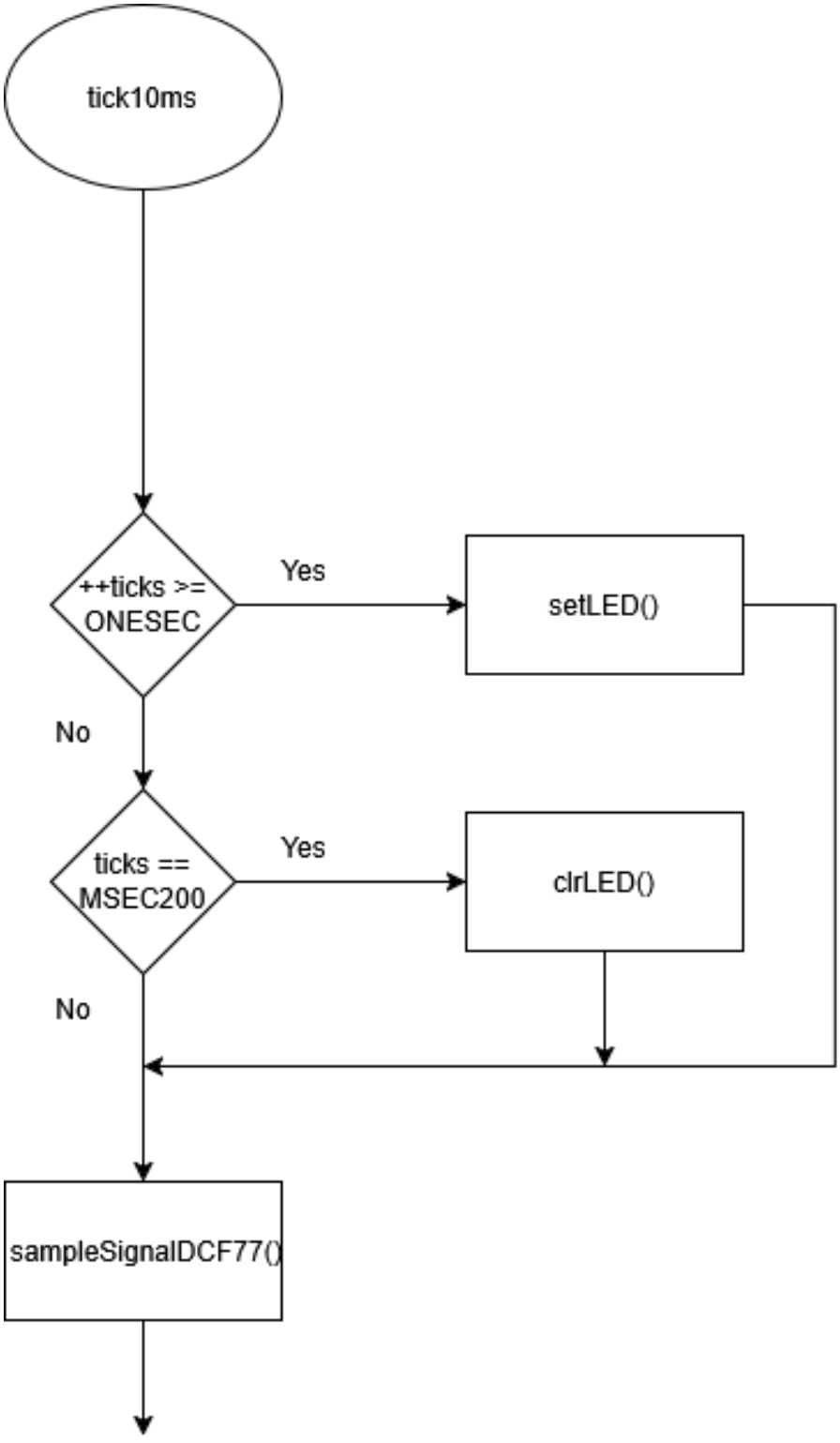

Figure 4: Clock 2

7.3 DCF77

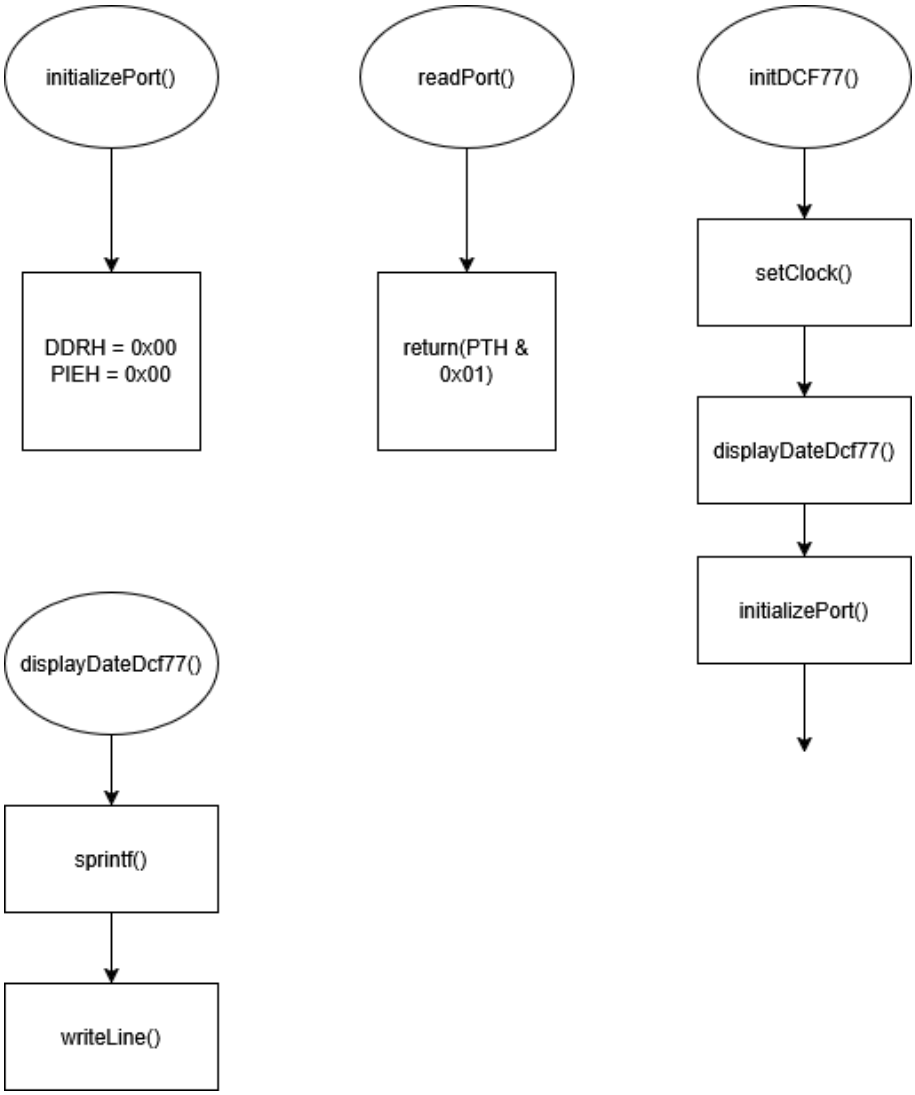

Figure 5: DCF77 - Part 1

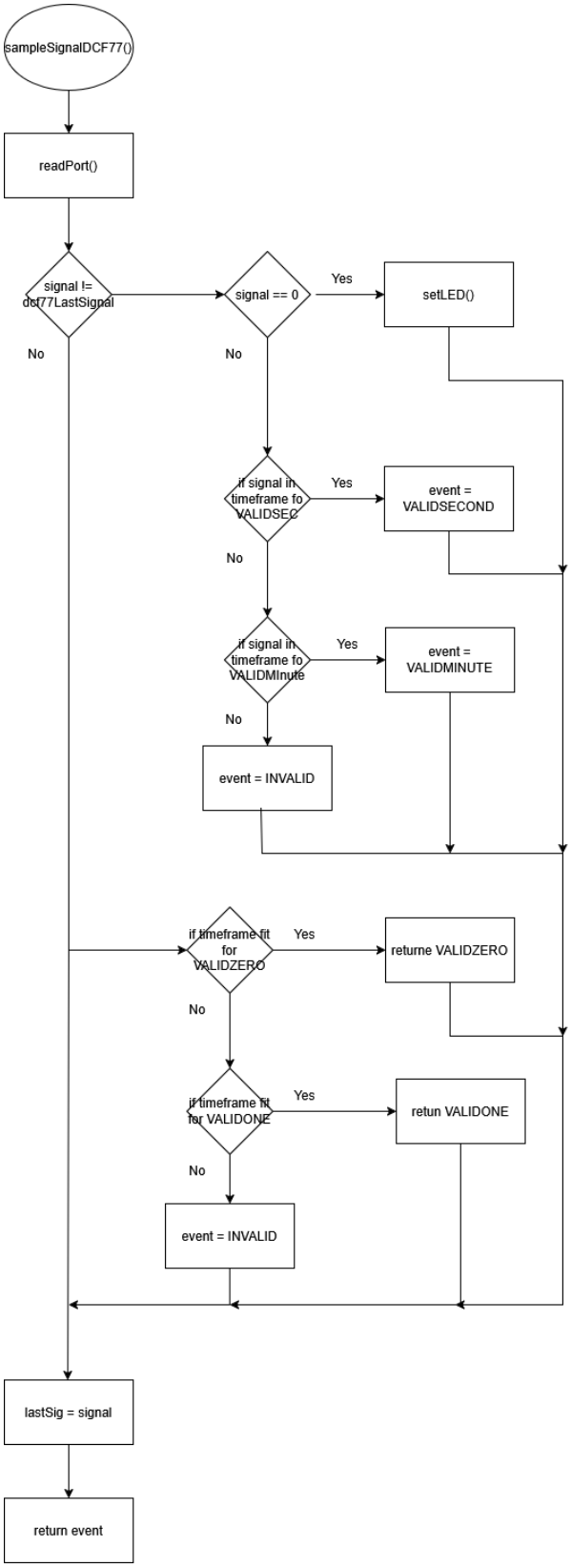

Figure 6: DCF77 - Part 2

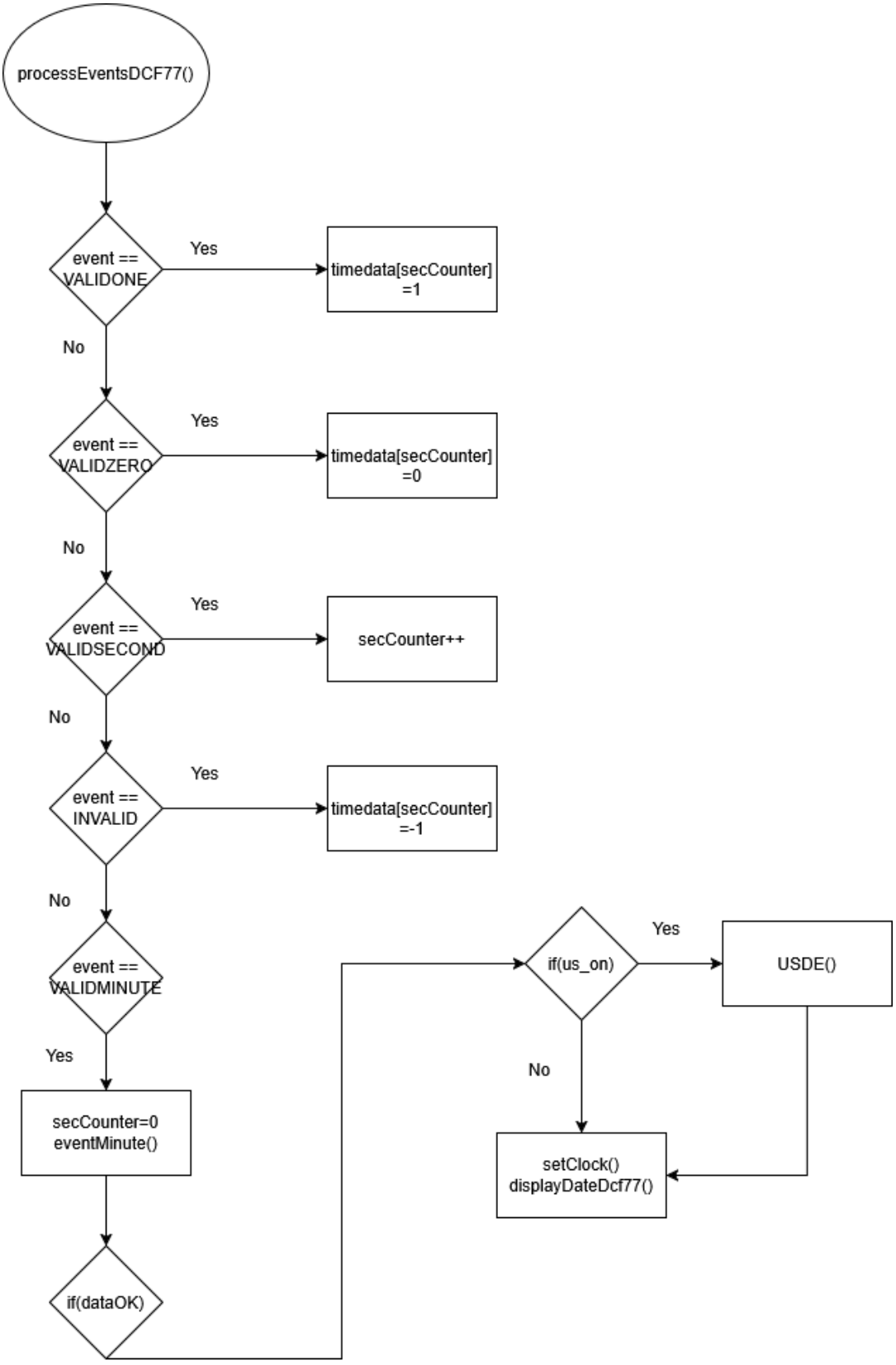

Figure 7: DCF77 - Part 3

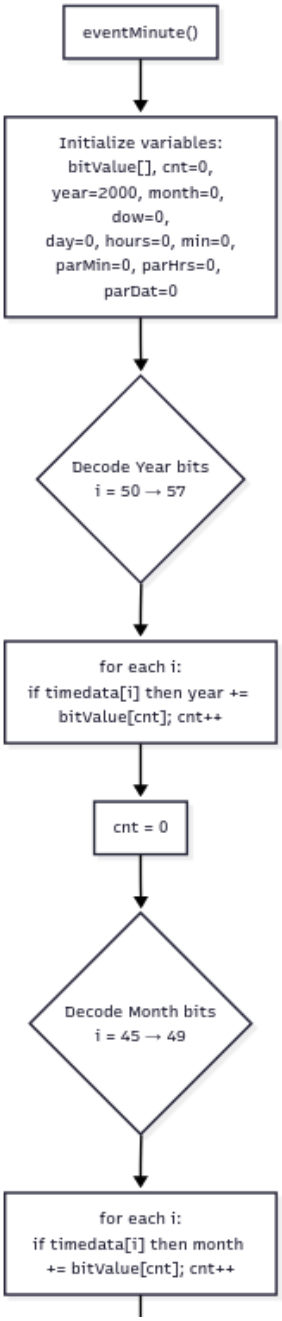

Figure 8: Event Minute – Part 1

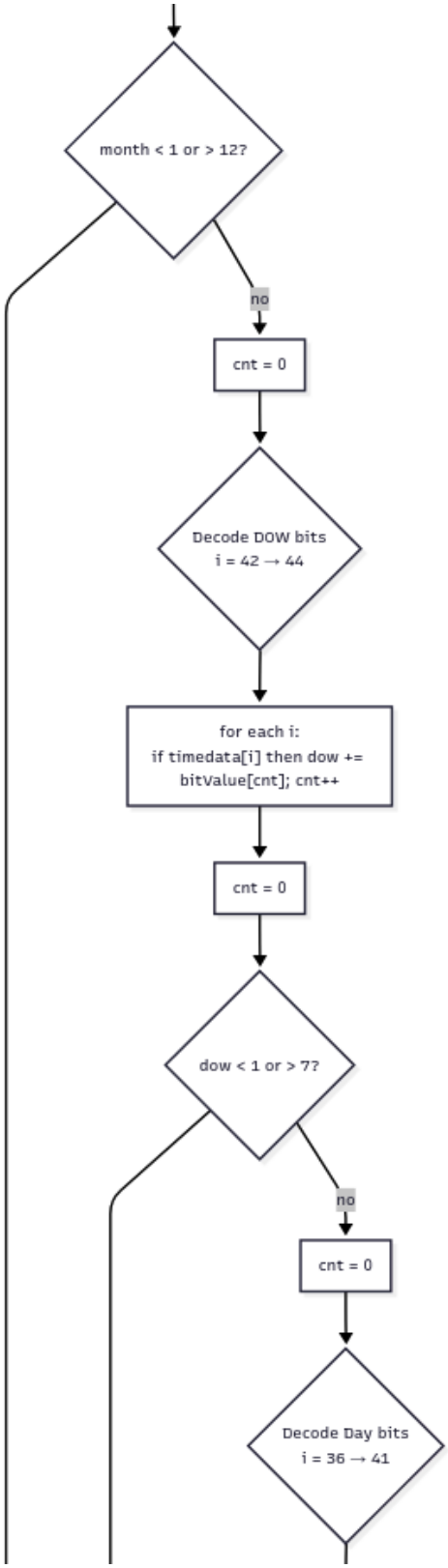

Figure 9: Event Minute – Part 2

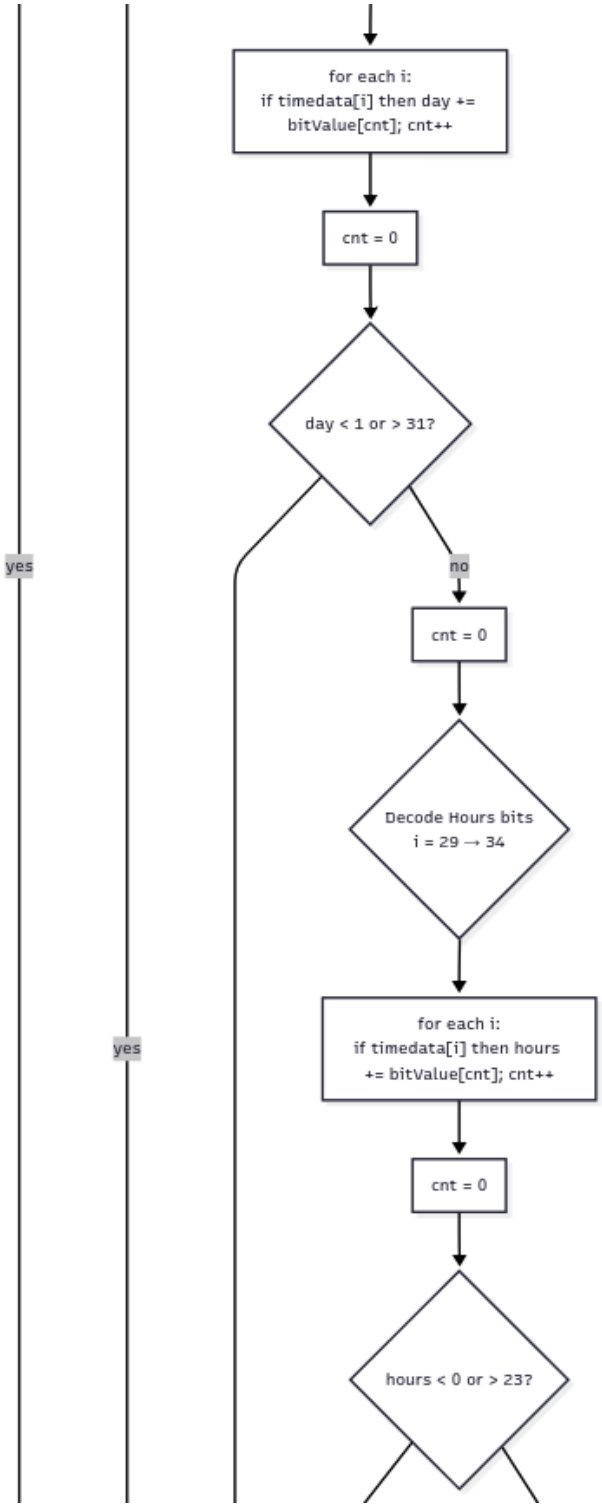

Figure 10: Event Minute – Part 3

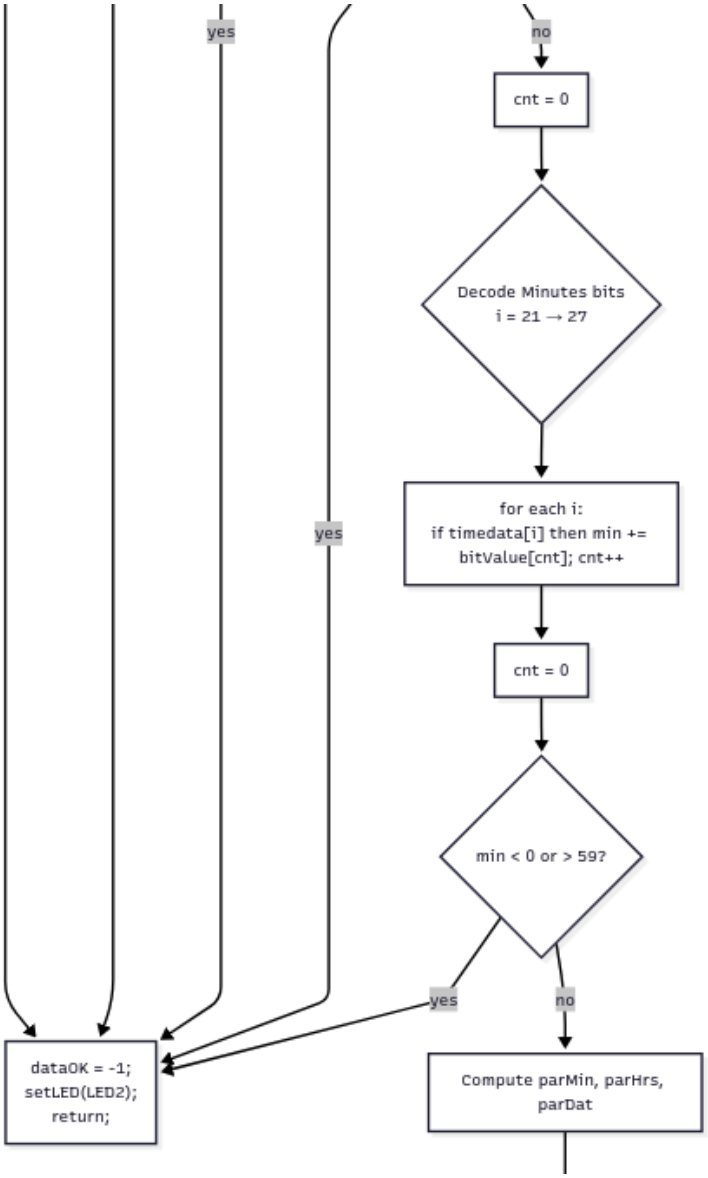

Figure 11: Event Minute – Part 4

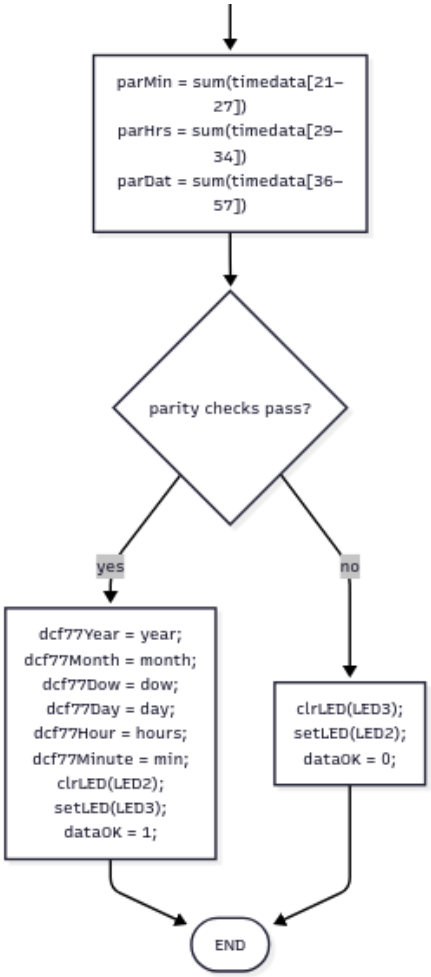

Figure 12: Event Minute – Part 5

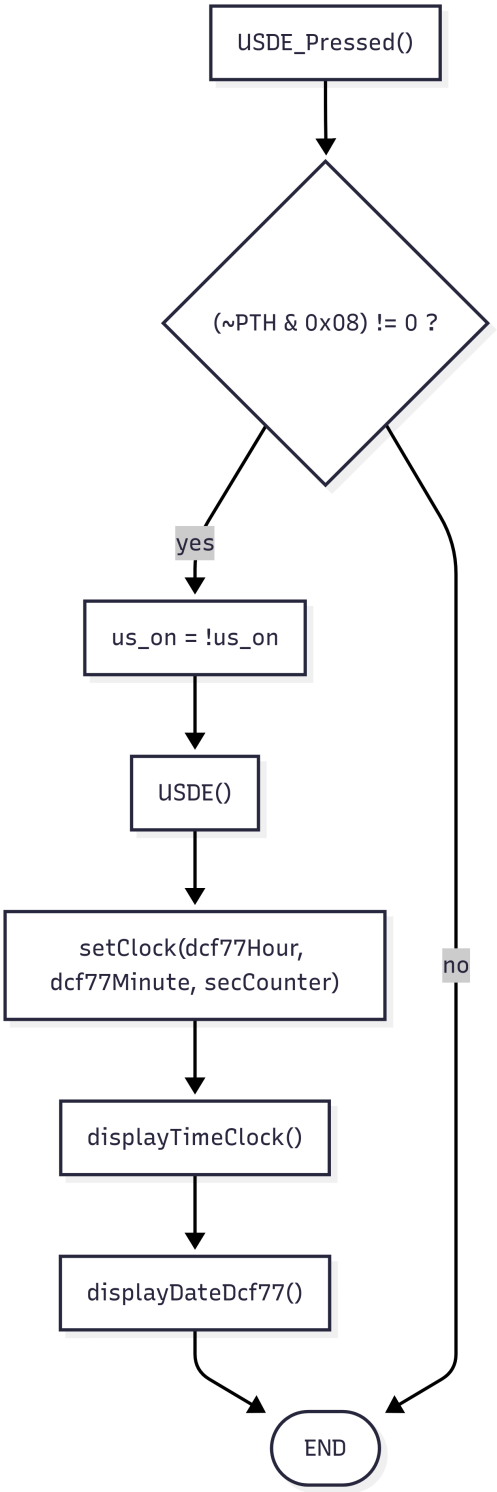

Figure 13: USDE Pressed

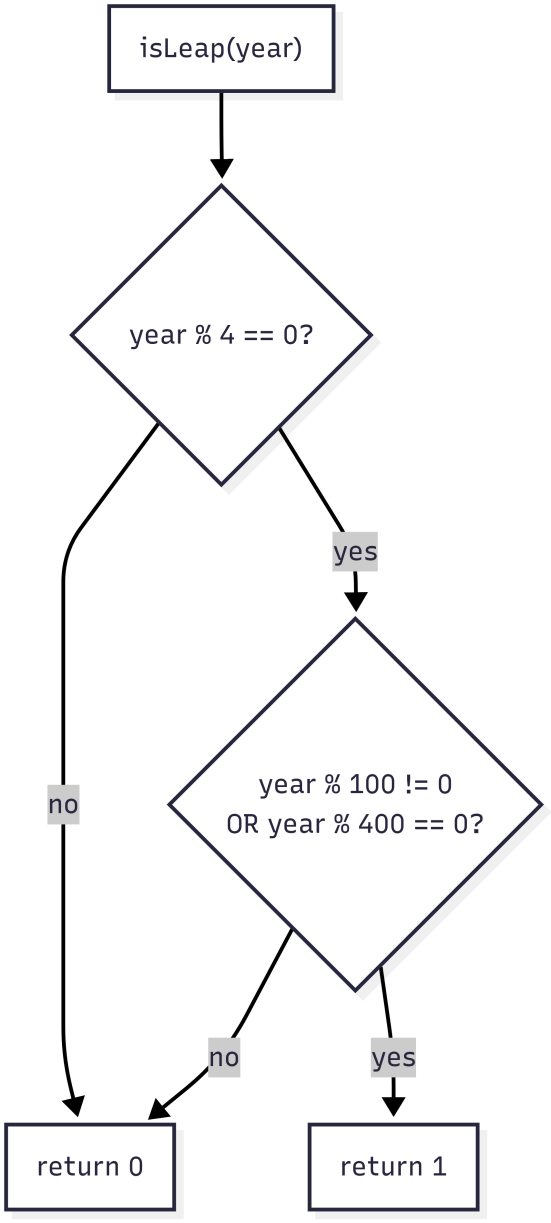

Figure 14: isLeap

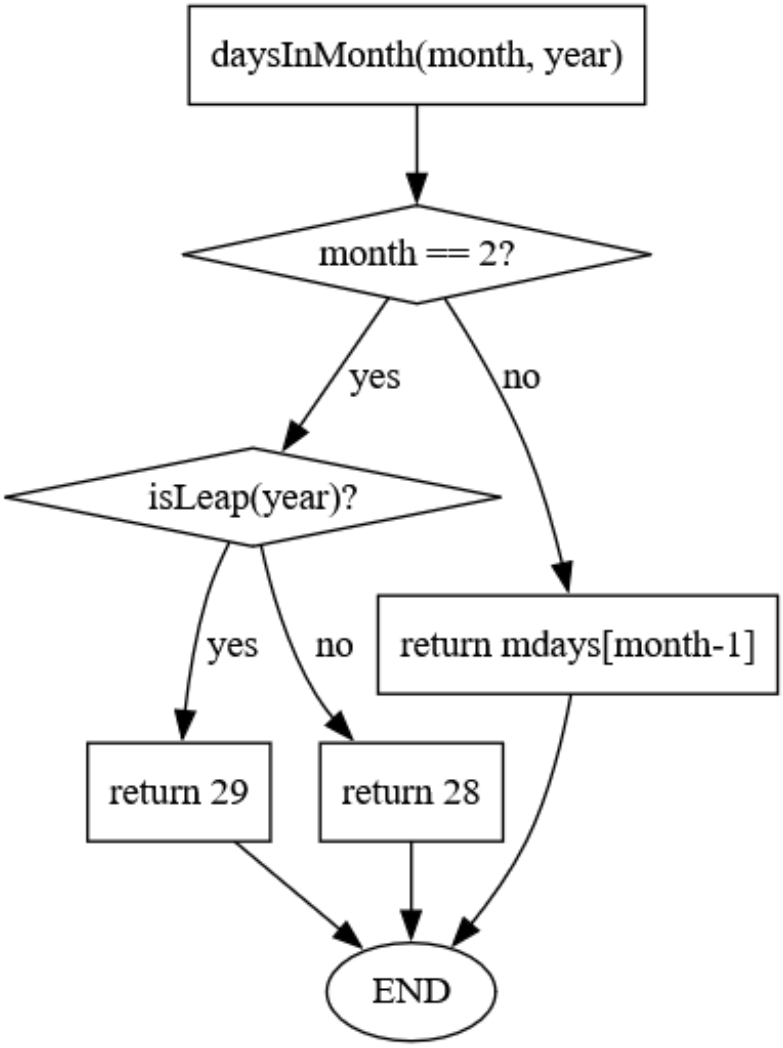

Figure 15: daysInMonth

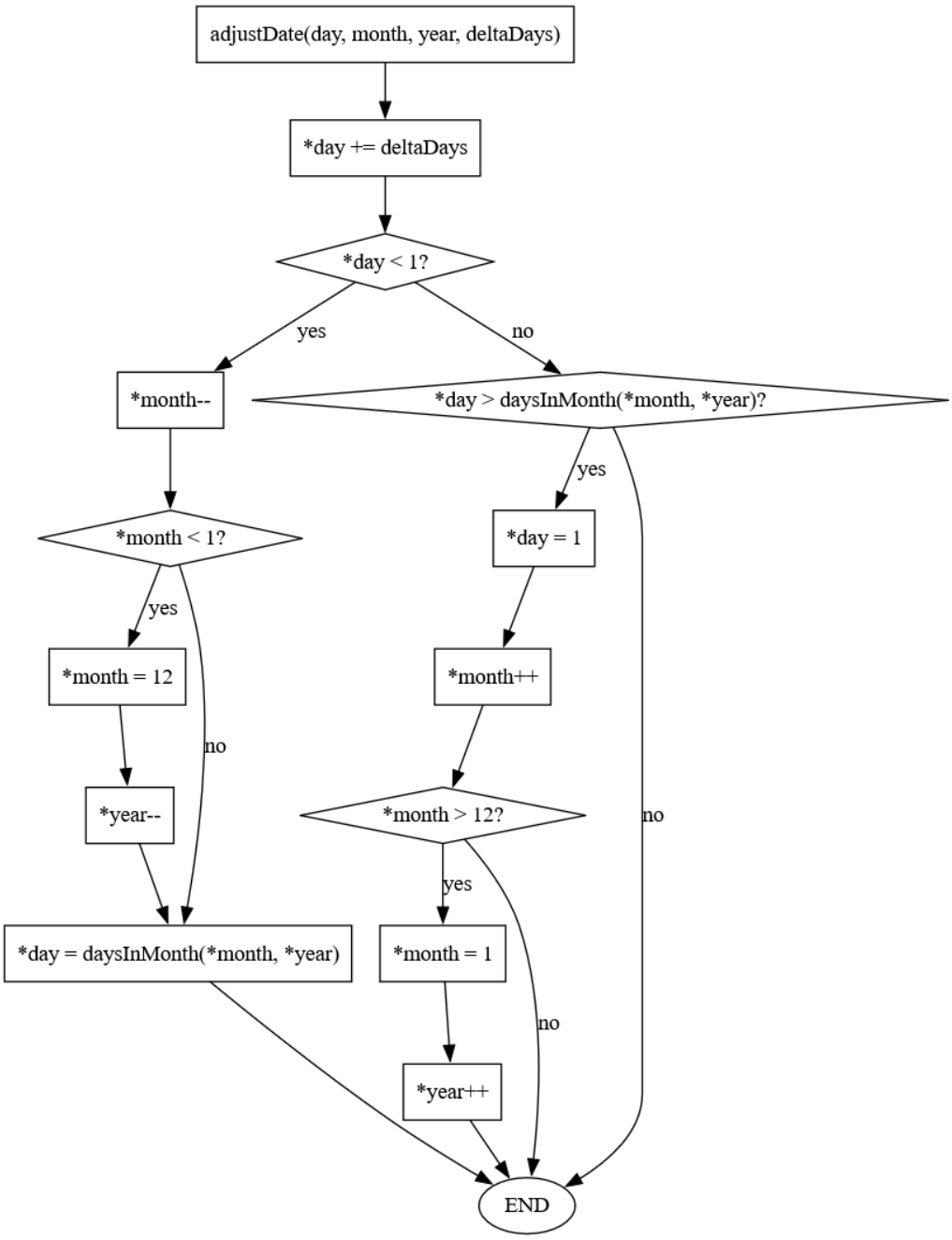

Figure 16: adjustDate

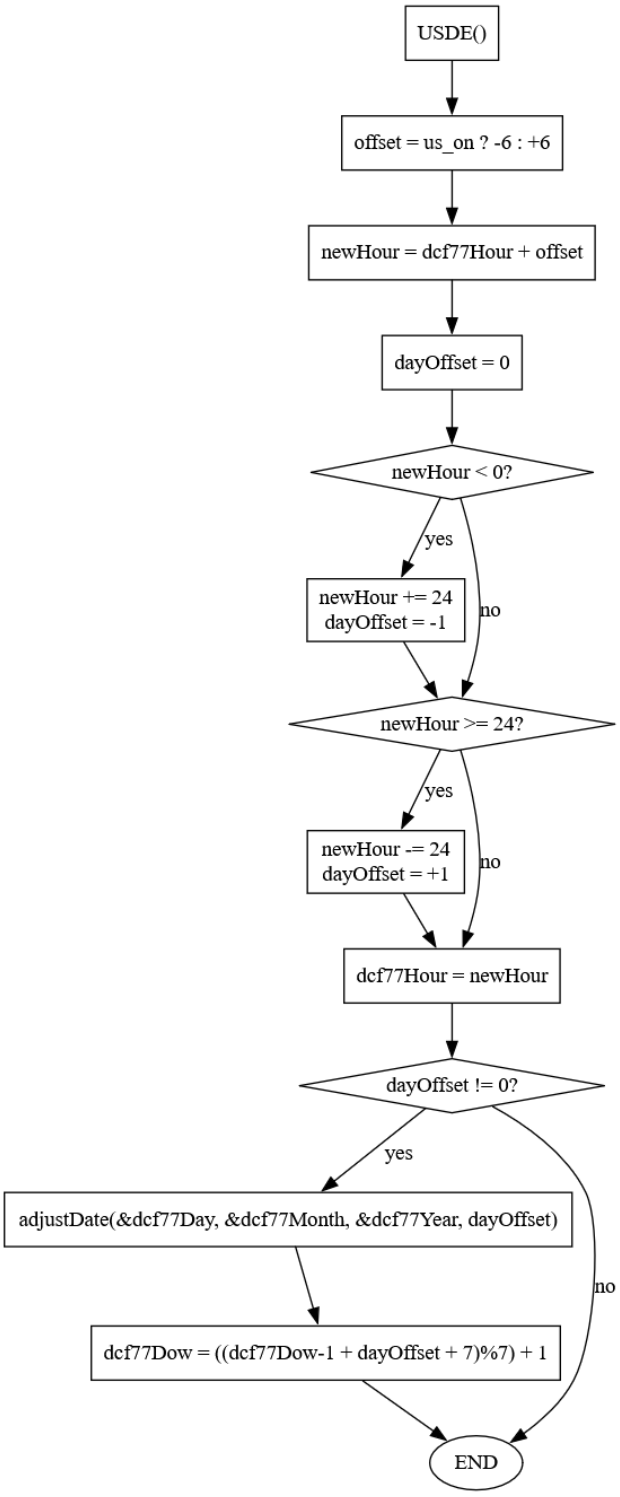

Figure 17: USDE