```
START

# Setup Libraries and Constants
INCLUDE necessary libraries for SPI, I2C, OLED Display, OneWire, and Dallas
Temperature
DEFINE constants:
    - PIN numbers for rotary encoder (CLK_PIN, DT_PIN, SW_PIN), Peltier control,
and I2C address
    - Temperature range limits (min_temperature, max_temperature)
    - PID controller gains (kp, ki, kd)
    - OLED screen dimensions and address

# Initialize Global Variables
INITIALIZE:
    - Screen index (scrn_index) to track which screen is active
    - Button press status (button_press) and toggle state for rotary encoder
(toggle)
    - Default target temperature (17) and variables for current temperature
    - PID control variables (integral, previous error, etc.)
    - EEPROM array to store previous target temperatures

# Setup Function
BEGIN setup:
    START serial communication for debugging
    INITIALIZE display and clear it
    INITIALIZE and start temperature sensor
    CONFIGURE pins for inputs (rotary encoder, button) and outputs (Peltier
control)
    ATTACH interrupts:
        - `rotary()` for handling rotary encoder rotations
        - `rotary_sw()` for handling button press detection
END setup

# Main Program Loop
BEGIN loop:
    WHILE true:
        CHECK the current screen index (scrn_index):
            IF screen index is 0:
                CALL screen_0()  # Main temperature display and control screen
            ELSE IF screen index is 1:
                CALL screen_1()  # Target temperature selection screen
            ELSE IF screen index is 2:
                CALL screen_2()  # Set new target temperature screen
            ELSE IF screen index is 3:
                CALL screen_3()  # View and set previous target temperatures
screen
END loop

# Helper Functions

FUNCTION array_append(new_temperature):
    FOR i from n-1 down to 1:
        SHIFT previous temperatures in the array to the right
    SET the first element of the array to new_temperature
    INCREMENT EEPROM data count if it’s not full
END function

FUNCTION rotary():
    READ encoder state
    IF encoder rotated (detect rising or falling edges):
        IF rotation is clockwise:
            IF scrn_index == 2:
                INCREASE target temperature (up to max_temperature)
            IF scrn_index == 3:
```

```
                INCREMENT EEPROM selection
        ELSE IF rotation is counterclockwise:
            IF scrn_index == 2:
                DECREASE target temperature (down to min_temperature)
            IF scrn_index == 3:
                DECREMENT EEPROM selection
    TOGGLE display update flag to refresh display
END function

FUNCTION rotary_sw():
    TOGGLE button press state (sw_flag) and set button_press flag
    WAIT briefly to debounce button
END function

FUNCTION read_temperature():
    REQUEST temperature from sensor
    IF reading is successful:
        RETURN current temperature
    ELSE:
        PRINT error message
        RETURN error code or default value
END function

# Screen Functions

FUNCTION screen_0():
    READ current temperature
    CALCULATE pid_output using PID control formula
    MAP pid_output to appropriate PWM values for Peltier control
    WRITE pid_output to Peltier output pin
    DISPLAY:
        - Target temperature
        - Current temperature
    IF button is pressed:
        SET screen index to 1 (screen_1)
        RESET button press flag
END function

FUNCTION screen_1():
    IF display update is needed:
        DISPLAY options to select new target or previous target temperature
    IF button is pressed:
        IF "new target" is selected:
            SET screen index to 2 (screen_2)
        ELSE:
            SET screen index to 3 (screen_3)
            RESET EEPROM selection to 1
        RESET button press flag
END function

FUNCTION screen_2():
    IF display update is needed:
        DISPLAY current target temperature setting
    IF button is pressed:
        CALL array_append to store target temperature in EEPROM
        DISPLAY confirmation of new target
        SET screen index to 0 (screen_0)
        RESET button press flag
END function

FUNCTION screen_3():
    IF EEPROM array is empty:
        SET screen index back to 0 (screen_0)
    ELSE:
```

```
        IF display update is needed:
            DISPLAY selected previous temperature from EEPROM array
        IF button is pressed:
            SET target_temperature to selected EEPROM value
            CALL array_append to store new target in EEPROM
            DISPLAY confirmation of target change
            SET screen index back to 0 (screen_0)
            RESET button press flag
END function

# PID Control Function

FUNCTION pid(error, kp, ki, kd):
    CALCULATE:
        - Proportional term as `proportional = error`
        - Integral term by accumulating error over time
        - Derivative term as rate of change of error
    COMPUTE pid_output using PID formula:
        pid_output = kp * proportional + ki * integral + kd * derivative
    RETURN pid_output
END function

STOP
```