// Inclusion de librerias

#include <stdio.h>

#include "pico/stdlib.h"

#include "hardware/adc.h"

#include "lcd\_i2c.h"

#include <math.h>

// GPIO del ADC

#define ANALOG\_GPIO  27

// Canal del ADC

#define ANALOG\_CHANNEL  0

// Tiempo de refresco para el siete segmentos

#define SLEEP\_MS 10

// Tiempo entre conversiones

#define ADC\_DELAY\_MS  500

// Constante de proporcionalidad para el NTC

#define BETA  3950

// Resistencia a 25 grados del NTC

#define RT0   10000

// Resistencia en serie al NTC

#define RS    10000

// Variable para almacenar el resultado del ADC

uint16\_t adc\_value = 0;

// Variable para guardar el valor de temperatura

float temperatura = 0.0;

/\*

 \* @brief Callback para la interrupcion de timer

 \* @param t: puntero a repeating\_timer

 \*/

bool muestreo\_periodico(struct repeating\_timer \*t) {

  // Lectura analogica (variable adc\_value)

  uint16\_t adc\_value = adc\_read();

  // Calcular valor de temperatura (variable temperatura)

  float R = (float)RS \* (1023.0 / (float)adc\_value - 1.0);

  temperatura = 1 / (log(1 / (4095. / adc\_value - 1)) / BETA + 1.0 / 298.15) - 273.15;

  return true;

}

/\*

 \* @brief Display temperature value in 7 segment display

 \* @param temperature: value to display

 \*/

void display\_temp(float temperatura) {

  // Variable para armar el string

  char str[16];

  // Armo string con temperatura

  sprintf(str, "Temp=%.2f C", temperatura);

  // Limpio display (si lo hubiese)

  lcd\_clear();

  // Muestro (printf o lcd\_string)

  lcd\_string(str);

}

/\*

 \* @brief Programa principal

 \*/

int main() {

  // Inicializacion de UART

  stdio\_init\_all();

  // Creo un repeating timer

  struct repeating\_timer timer;

  // Creo un callback para la interrupcion del timer

  add\_repeating\_timer\_ms(ADC\_DELAY\_MS, muestreo\_periodico, NULL, &timer);

  // Inicializo ADC

  adc\_init();

  // Inicializo GPIO27 como entrada analogica

  adc\_gpio\_init(27);

  // Selecciono canal analogico

  adc\_select\_input(1);

  // Solo si se usa LCD con I2C

  // Configuro el I2C0 a 100 KHz de clock

  i2c\_init(i2c0, 100 \* 1000);

  // Elijo GPIO4 como linea de SDA

  gpio\_set\_function(4, GPIO\_FUNC\_I2C);

  // Elijo GPIO5 como linea de SCL

  gpio\_set\_function(5, GPIO\_FUNC\_I2C);

  // Activo pull-up en ambos GPIO, son debiles por lo que

  // es recomendable usar pull-ups externas

  gpio\_pull\_up(4);

  gpio\_pull\_up(5);

  // Inicializo display (si lo hubiese)

  lcd\_init();

  // Bucle infinito

  while (true) {

    // Muestro temperatura

    display\_temp(temperatura);

    // Espero

    sleep\_ms(1500);

  }

}