Temperature Meter Using DS18B20 Temperature Sensor and ESP32

Members:
Yogesh Deshpande - 181070074
Archeel Parekh - 181070044
Srushti Shah - 191071902
Manasi Deshmukh- 191071904

SUBJECT: Digital Logic Design

AIM:

To make a temperature meter using basic electrical and electronic components.

APPARATUS:

1. ESP32
2. DS18B20 temperature sensor
3. 4.7K resistor
4. Jumper wires
5. Breadboard

THEORY:

ESP32 is a series of low-cost, low-power system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth.

ESP32 Technical Specifications

Microprocessor	Tensilica Xtensa LX6
Maximum Operating Frequency	240MHz
Operating Voltage	3.3V
Analog Input Pins	12-bit, 18 Channel
DAC Pins	8-bit, 2 Channel
Digital I/O Pins	39 (of which 34 is normal GPIO pin)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
SRAM	520 KB
Communication	SPI(4), I2C(2), I2S(2), CAN, UART(3)
Wi-Fi	802.11 b/g/n
Bluetooth	V4.2 – Supports BLE and Classic Bluetooth

Pin Desciption for ESP32

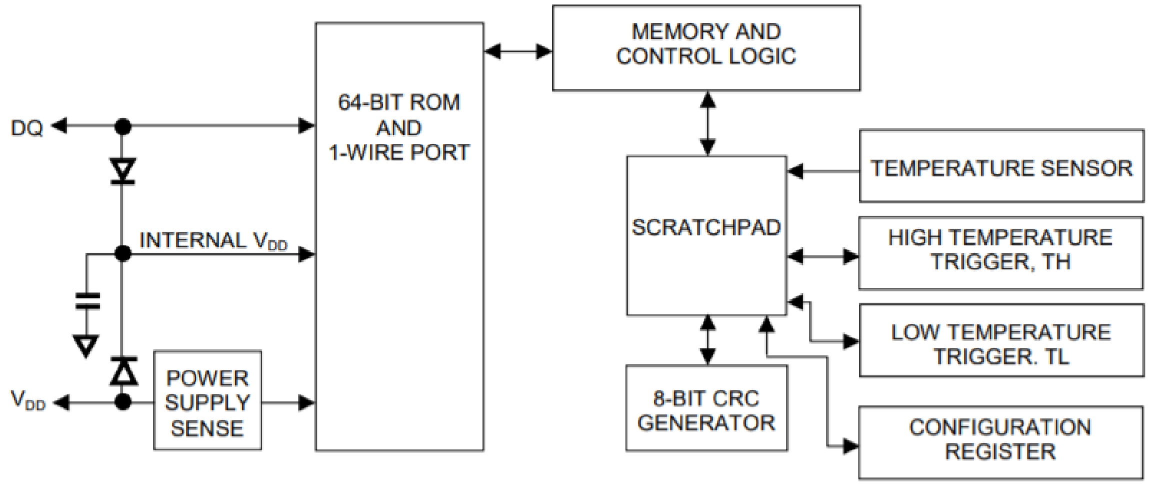
Pin Category	Pin Name	Details
Power	Micro-USB, 3.3V, 5V, GND	Micro-USB: ESP32 can be powered through USB port 5V: Regulated 5V can be supplied to this pin which is we be again regulated to 3.3V by on board regulator, to power the board. 3.3V: Regulated 3.3V can be supplied to this pin to power the board. GND: Ground pins.
Enable	En	The pin and the button resets the microcontroller.
Analog Pins	ADC1_0 to ADC1_5 and ADC2_0 to ADC2_9	Used to measure analog voltage in the range of 0-3.3V. 12-bit 18 Channel ADC

DAC pins	DAC1 and DAC2	Used for Digital to analog Conversion
Input/Output Pins	GPIO0 to GPIO39	Totally 39 GPIO pins, can be used as input or output pins. 0V (low) and 3.3V (high). But pins 34 to 39 can be used as input only
Capacitive Touch pins	T0 to T9	These 10 pins can be used a touch pins normally used for capacitive pads
RTC GPIO pins	RTCIO0 to RTCIO17	These 18 GPIO pins can be used to wake up the ESP32 from deep sleep mode.
Serial	Rx, Tx	Used to receive and transmit TTL serial data.
External Interrupts	All GPIO	Any GPIO can be use to trigger an interrupt.
PWM	All GPIO	16 independent channel is available for PWM any GPIO can be made to work as PWM though software
VSPI	GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS)	Used for SPI-1 communication.
HSPI	GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS)	Used for SPI-2 communication.
IIC	GPIO21(SDA), GPIO22(SCL)	Used for I2C communication.
AREF	AREF	To provide reference voltage for input voltage.

The DS18B20 temperature is a 1-wire digital temperature sensor with the following features:

- Power supply range: 3.0V to 5.5V
- Communicates over 1-wire bus communication
- Operating range temperature: -55ºC to 125ºC
- Accuracy +/-0.5 ºC (between the range -10ºC to 85ºC)

Block diagram of DS18B20 internals:



The ESP32 in combination with DS18B20 temperature sensor will be used to make the temperature meter.

Arduino IDE libraries used:

- OneWire library by Paul Stoffregen
- Dallas Temperature library

The temperature sensor is connected to GPIO 4.

PROGRAM:

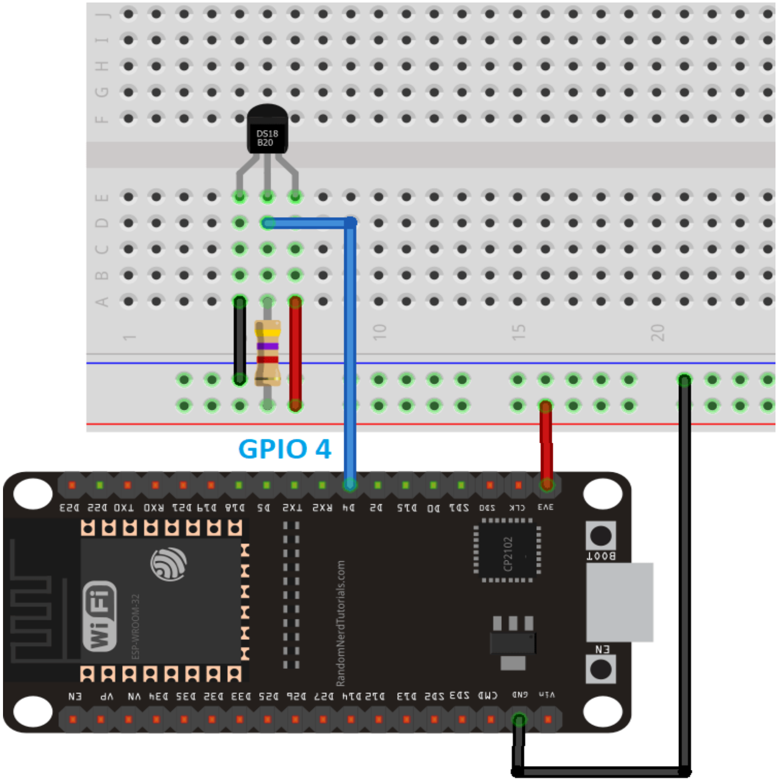
Code flashed onto ESP32:

```
#include <OneWire.h>
#include <DallasTemperature.h>

// GPIO where the DS18B20 is connected to
const int oneWireBus = 4;

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(oneWireBus);

// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);

void setup() {
   // Start the Serial Monitor
   Serial.begin(115200);
   // Start the DS18B20 sensor
   sensors.begin();
}

void loop() {
   sensors.requestTemperatures();
```

```
  float temperatureC = sensors.getTempCByIndex(0);
  float temperatureF = sensors.getTempFByIndex(0);
  Serial.print(temperatureC);
  Serial.println("ºC");
  Serial.print(temperatureF);
  Serial.println("ºF");
  delay(5000);
}
```

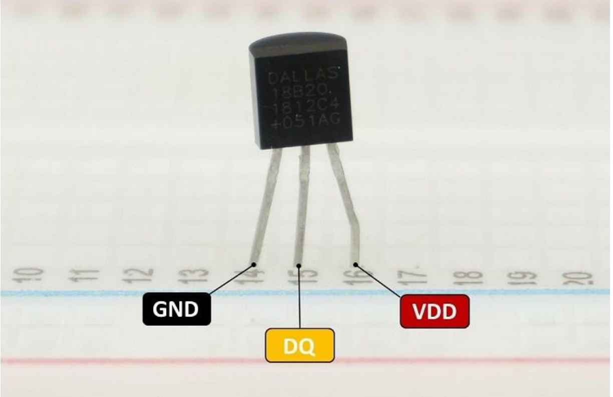
SCHEMATIC DIAGRAM:

(Normal mode - ESP32 powered through V_{DD} pin)



WORKING:

The DS18B20 temperature sensor is a one-wire digital temperature sensor. This means that it just requires one data line (and GND) to communicate with your ESP32. It can be powered by an external power supply or it can derive power from the data line (called “parasite mode”), which eliminates the need for an external power supply.



Each DS18B20 temperature sensor has a unique 64-bit serial code. This allows you to wire multiple sensors to the same data wire. So, you can get temperature from multiple sensors using just one GPIO.

Normal (external supply) mode:

With an external supply, three wires are required: the bus wire, ground, and power. The 4.7k pull-up resistor is still required on the bus wire. As the bus is free for data transfer, the microcontroller can continually poll the state of a device doing a conversion. This way, a conversion request can finish as soon as the device reports being done, as opposed to having to wait for conversion time (dependent on device function and resolution) in "parasite" power mode.

Wiring:

- The GND pin of the DS18B20 goes to GND
- The Vdd pin of the DS18B20 goes to +5V
- The Data pin of the DS18B20 goes to a (digital) pin GPIO 4

FUTURE SCOPE:

- Usage of multiple DS18B20 temperature sensors for more accurate readings.
- Different compact design.
- Adding a digital display to view the temperature.

CONCLUSION:

Thus, we have developed a temperature sensor using ESP32 and DS18B20 temperature sensor and OneWire Arduino IDE library. This product measures the current temperature of the surrounding.