

Assignment 2: Reading temperature sensor: RPi transactions with DHT11 sensor

DHT11 is an intelligent temperature/humidity sensor which according to its specs can capture accurate temperature and humidity environmental data. In this assignment, you will attach the 3 pins of DHT11 to the RPi's GPIOs, for Power, Ground and data connectivity. The communication between the DHT11 and RPi is asynchronous performed on a single bus line, without clocks. DHT11 processes its raw analog data into digital and then packages every reading into a 5-byte data format. Every RPi transaction entails capturing the data the 40-bit format from the DHT11 over the bus line. Two important points: a) every the RPi DHT11 transaction is based on a request/acknowledge type of protocol, b) communication takes place bit-serially where the information bits are dispatched in sequences of pulses of specific time duration. RPi needs to interpret these timing pulses after capturing them.

Details of the operation and transaction protocol of DHT11 are provided separately. You will need to use the timing functions of the wiringPi library.

Part 1

Use the polling I/O technique to implement in C one RPi DHT11 transaction. You should verify the transaction output using its checksum data. You should also return the temperature and humidity output, print it on the screen and save it in a file (include date/time.)

Part 2

Repeat Part 1 using the edge-triggered interrupt technique. Comment on the differences, if any.

Part 3

Write a loop of your C code to record temperature data for a short time. Connect your LED circuit from Assignment 1 so that if any of the temperatures goes above a threshold, RPi will signal the LED to turn on. To force such an event, we could place a soldering iron tip close enough (but not too close) to the DHT11 sensor.

Let's see how it works.

Part 4

In this Part of Assignment 2, we want to use a Display to show the temperature or humidity once it is recorded by RPi. Connect the DISPLAY board to be provided to each group on the RPi GPIOs. Develop a C or Python routine to show the RPi transaction data on the display. Details about the Display will be provided.