Lab text yazısı plagarism vs

Start project to build a flashing LED (ATh00)

Our first code for A-Uno is named ATh00. It should implement the flashing of the internal LED by the described FSM. To create an Arduino firmware project, we follow the given instructions such as:

(New project) Name ATh00; Path work folder; New Project, Next (Schematics Design) Create a schematics, Next

(PCB layout) Do not create a PCB Layout, Next

(Firmware) Create Firmware Project, Family ARDUINO;

Controller: Arduino Uno; Compiler Arduino AVR (Proteus);

Create Quick Start Files: yes; Create Peripherals: yes; Next and Finish.

Then, we have a Schematic Capture and a Source Code page in Proteus. The schematic page contains an Arduino-Uno block. Source Code contains a coding template.

Design of flashing LED (ATh01)

The internal LED of Arduino is planned to be used for system-ready

indication by flashing it once every two seconds for a 0.2s period. We will do it by using 0.01s (10ms) trigger periods for the time base of our

design.

FSM of the flashing led gelecek

Kodları gelecek

delay(10): Delays the execution of the code for 10 milliseconds.

flashState: Keeps track of the current state of a flashing LED state machine.

flashCount: Counts the number of iterations or flashes before changing the state.

digitalWrite(13, HIGH): Sets the digital pin 13 (typically connected to an LED) to a HIGH state, turning the LED on.

digitalWrite(13, LOW): Sets the digital pin 13 to a LOW state, turning the LED off.

4.3.5 Flashing LED Test procedures

To observe the LED-on period and total period of the internal LED we will use an oscilloscope and record the readings.

The timing of the LED is measured using an oscilloscope. The oscilloscope is set to horizontal: ramp, pos=200, sweep=0.2s/div; trigger: ch-A, Level 8, DC, rising-edge, Auto, cursors; Ch-A: Pos 120, DC, 2V/div. Cursors placed to measure the periods of high=200ms, and low=1.80s by us.

The test ended successfully.

Details of the oscilloscope can be found in the appendix section.

4.3.6 Answer these questions

7 soru

4.4 Incremental Design for Serial Debug Port (ATh02)

kod gelecek

In that task, we need the virtual terminal. By following the documentation we prepared the circuit. The code snippet "Serial.print('+'); " has been added to flashState == 1 code block for the task. After the required modification, we can test it.

In the test, the value you set, and you measured are different values. The test aims to measure the value and compare it against the expected one.

4 soru

4.5 AD Converter test (ATh03).

Again, the documentation was followed by us to satisfy the required task. We implemented the desired hardware and software on both sides. Also, we run the simulation on Scilab and the diagrams can be found in the appendix section.

7 soru

4.6 Heater FSM test (ATh04)

We did the necessary modifications for the board and code sections. Defined new variables like htrState, Hoff, HOn, etc. Inserted the pinMode (12, OUTPUT ); in the setup block for the heater.

After running it, we see that the flashing LED is still working as expected, the ADC reading and the calculated temperature are displayed when the sensor temperature changes, "+" is displayed, and the heater-LED turns on when T<18C, and "-" is displayed, and heater-LED turns off when T>22. Any contradiction between the sensor temperature and the calculated temperature can be solved by Tm = ADRes \* 0.1 code modification.

4 soru (FSM çiz, kodu elle yaz, kodu elle yaz, code modification for the loop() section can be found in the appendix)

4.7 NTC thermistor as temperature Sensor

In that task, we will be using an NTC sensor circuit to measure the performance of the sensor. Again made some modifications in both board and code sections.

For all temperatures from 0 to 50C, we plot the measured temperature Tm versus the entered sensor temperature Ts curve by using SciLab.

If we delete the plot and then write other SciLab commands we obtain the second plot instead of the combination of two plots. These plots give the linearity and error.

4.8 Conclusion

This report has addressed several embedded systems-related topics, such as creating and testing a flashing LED, debugging code using a serial debug connection, examining sensor linearity and ADC output, and testing a heater FSM. We have spoken about how the flashing LED is designed and how to test it to make sure it works. We also looked at how to create code debugging using a serial debug port. To guarantee an exact match between the measured and actual temperatures, we also examined the ADC output and the linearity of the sensor and made adjustments. Lastly, we performed a heater FSM test and talked about the adjustments needed to get it to start in the heating condition.