EMBEDDED SYSTEM DESIGN PROJECT

The aim of this project is to measure heartbeat with max30100 heartbeat and pulse oximetry sensor and to receive time information from the server with network time protocol using EK-TM4C1294XL launchpad. While doing all these, tasks are communicated by using semaphores, events and mailboxes that provide interprocess communication and deadlocks are prevented. Also, attention is paid not to mutual explusion. At the end of the process, network communication is provided between the tcp server with the specific IP address and port number with the help of Hercules program.

The max30100 sensor is a digital output sensor that communicates with the i2c protocol. In my project, I received and wrote data by communicating with the necessary register addresses, using the i2c read-write functions, the information I obtained from the datasheet of the MAX30100 sensor.

The MAX30100 is an integrated pulse oximetry and heart-rate monitor sensor.

It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals.

The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

I placed the sensor on my EK-TM4C1294XL launchpad together with the Sensor Hub Booster Pack we have, since there was a design flaw with the pull up resistors on the max30100.

In order to receive data with the max30100 sensor, there are registers that need to be set first. These registers and their operations are as follows:

Set mode

Set sample rates

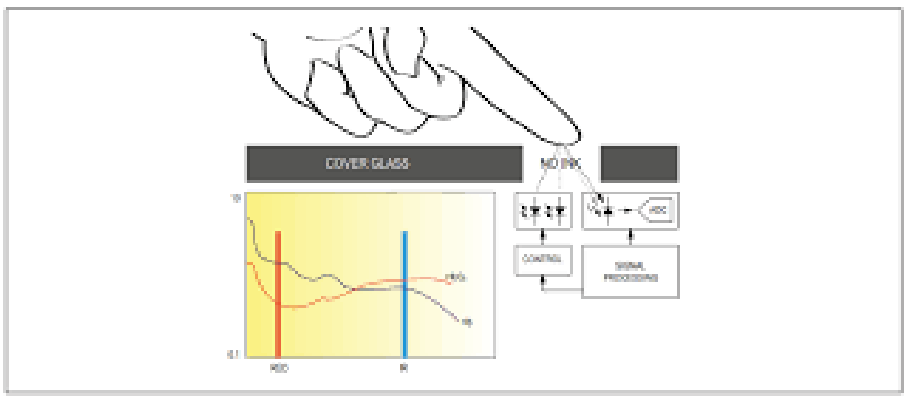
Set led currents

Set led pulse widths

Enable interrupts

Since the values in all these registers change according to the operations performed, at the very beginning of the main function, fifo registers, interrupts, led currents and mode configuration are reset.

System Block Diagram



After all these register adjustments, reading is done from fifo data in the while loop of our main task. This reading is not like reading other registers because we will take 4 bytes and FIFO data does not repeat its address during register reading. Therefore, I made this reading with the function called readFifoData, not with the readReg function.

Although this function is similar to the read Reg function, it is different with the for loop inside.

After reading, the heart rate that should be 130 with dc offset within this value is based on 60 thousand. so I applied a dc filter to remove the DC offset. However, although this filter is not sufficient, it is also applied in a filter called Mean filter and butterworth. After all this, I divided the data I got into 1200, taking into account this cycle that takes 50 ns pause with task sleep and a minute is 60 seconds. I now transfer this heart rate from i2c to the tcp server with the serverSocketTask I created. Communication between two servers is provided by sending messages such as HEARTRATE, AGE from the Hercules program and normal heart rate monitor works with HEARTRATE. With the AGE command, the age is entered later, and the age and heart rate data sent to the ageTask via mailboxes, and the heart rate data suitable for the age of the person are displayed.

Halil İbrahim ÖZTÜRK