

# Designing of Sleep Quality Meter

Suneetha Uppala, K.Gotuhami

**Abstract**—A sleep quality meter that uses to judge the quality of the user's sleep. A system that monitors the quality of the sleep of its user. This project could be used to determine the factors that lead to an optimal sleep at an individual level. Maybe This project aims to provide a concrete metric for the quality of sleep. The project consists of two main pieces a receiving unit and a sensor unit. The sensor unit contains a heartbeat sensor, a gyroscope and an accelerometer to measure movement, a conductive stretchy cord to measure breathing rate, and a wireless transceiver to relay all of this information to the receiving unit. The receiving unit receives the data from the sensor unit, analyzes it and saves so that the user can look on the quality of their sleep.

**Keywords**— Sleep quality, sensor unit, receiving unit and Nrf wireless data transceiver ect

## I. INTRODUCTION

The idea for this project came from biomedical interests[1]. Sleep is a mysterious thing, something that we can't reflect on or really improve on without outside aid. Solving sleeping problems is a pretty large industry, from various sleep monitoring apps, to sleeping pills, all aimed at helping their user sleep better. Many of these apps work by placing a phone under your pillow before sleeping and it uses either the phone's microphone or the phone's accelerometer to try and detect movement throughout the night. We thought that combining this idea with a little extra hardware might give us the ability to get access to more metrics during sleep, which might give us better insights into what makes good and bad sleep for us personally.

The metrics we chose to focus on are heart rate, breathing rate, and movement. We believe that these three metrics are a good representation of a person's restfulness during sleep. Through some preliminary research, we discovered that the differences between sleep states are mostly discerned through changes in bodily functions. When we are awake out heartrate, breathing, and brain wave activity is very irregular. When we fall asleep there are two main sleep patterns: REM and non-REM sleep. In non-REM sleep, our heart rate and breathing decreases and becomes quite regular. However, when we are in REM sleep our vitals are almost just as irregular as when we are awake except for movement. During REM we are temporarily paralyzed, so we shouldn't see any major movement[2].

### Hardware Details:

The project consists of two main pieces: the sensor unit and the receiving unit. The sensor unit is having different sensor sense the user as they sleep, and the receiving unit receives data in real time and computes various sleep metrics. The sensor unit records movement via an

accelerometer, heartrate and breathing rate via a stretchable conductive cord. Below is a block diagram is shown in figure 1.

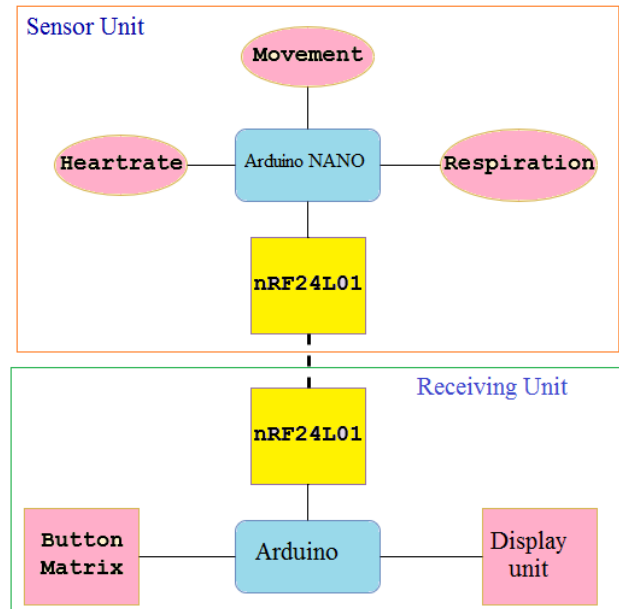


Figure 1: Block diagram of sleep quality meter

## 1. THE SENSOR UNIT

The sensor unit serves as a remote sensor worn by the user, relaying data to the receiving unit in realtime. It consists of an ArduinoNano, an acclerometer/gyroscope combined IC, an nRF24L01 2.4 GHz transceiver, circuitry for a breathing rate sensor, and circuitry for a heartbeatsensor. The connections for the various components on the sensor unit are fairly straightforward. The accelerometer and wireless transceiver both communicate with the Arduino Nano over SPI, and while the nRF24L01 is powered by a maximum of 3.6V, the data lines are all 5V tolerant, which makes direct wiring between the two possible. Theheartrate sensor consists of an IR transmitter, an IR phototransistor, OP amp, and filter circuitry. The IR LED emits infrared light into the user's finger, of which portions are reflected back out of the finger and hit the phototransistor. The amount reflected changes with blood pressure and blood oxygen level.

The output of the phototransistor is filtered through a high pass filter, and then gets amplified 200x by a non inverting amplifier. The output of this amplifier is read by the arduino.

Revised Manuscript Received on April 12, 2019.

Dr.SuneethaUppala, Dept. of Electronics, S.K.university, Anantapur,A.P,India.

Dr.K.Gotuhami, Department of Electronics,Malineni Lakshmaiah Women's Engineering, Guntur.

## Designing Of Sleep Quality Meter

In order to read the breathing rate of the user, we used a conductive stretchable cord that changes its resistance as it expands and contracts. We use this as the top half a voltage divider in order to sense the compression or expansion of the user's chest/abdomen, corresponding to when they are inhaling or exhaling. The output of the voltage divider is read by an ADC connected to the nano, whose readings are relayed to the receiving unit arduino for analysis. The sensor unit photo is shown in below photograph 1.

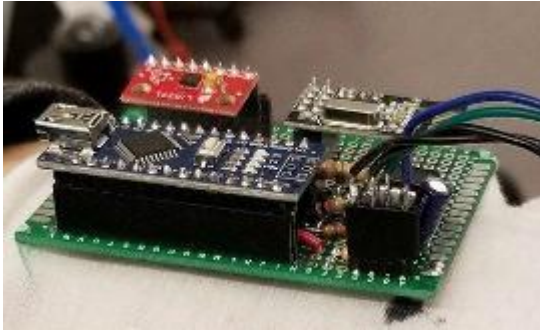


Photo graph 1: sensor unit of sleep quality meter.

### 1.1 Heart Rate

The heartbeat sensor we made performed quite well once we had it permanently mounted to a protoboard to keep a constant distance between the IR emitted and receiver. Its reliability increased even further when it was mounted inside the glove, providing a relatively constant pressure between the sensor and the user's finger. The sensor picks up the initial rise in blood pressure, the following dip, and then the following rise, corresponding to a single heartbeat:

### 1.2 Breathing Rate

In order to try and gain access to another metric to base our sleep metric on, we added a breathing rate sensor, with the main component being a conductive stretchable cord. The cord's resistance changes as it expands and contracts, and we can use this as part of a resistor divider to measure the change in voltage. Over time, we were able to get a pretty clear view of the rising and falling of a person's chest[3].

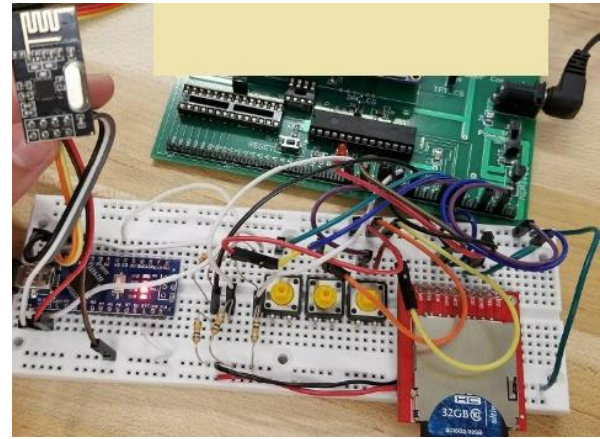
## 2. RESULTS & DISCUSSIONS

The receiving unit consists of Arduino, button matrix, display unit and wireless transceiver. When the sensor unit identified any sensor data which is sent through the NRF wireless sensors to receiving unit. This data is analysed in the receiving unit controller through code. This analysed data is displayed in the display unit. The matrix can switch the program according to the corresponding sensor data which is going to display in the display unit.

### 2.1 NRF 24L01

In our project, we utilized a module containing the nRF24L01 2.4 GHz wireless transceiver IC. This device operates in the 2.4 GHz range, in the band allocated for non-licensed use in the United States. Our device is compliant with a Class B transceiver according to Part 15 of Title 47 of the FCC rules (47 CFR 15) which states that the device may not cause harmful interference, and the device must accept any interference received, including interference that may

cause undesired operation. The photograph of receiving unit is shown in below photograph 2.



Photograph 2: receiving unit of sleep quality meter.

### Hardware/Software Tradeoffs

In this design there were a few trade-offs that we were faced with. We had to keep the hardware quite small, which limited the types of sensors that we could use, as well as forcing us to use the smallest components possible[4]. We also had to make sure each of the components on the sensor unit itself were relatively low power, so that it would last throughout a typical 8 hour night's sleep and easily warble.

In order to communicate with the Arduino Nano, we decided it would be best to connect the two over SPI (see hardware section for more details). Since the Arduino is frequently communicating with its wireless transceiver and the receiving is only seldom communicating with the SD card (in large bursts), we kept the Arduino (sensor unit) as the SPI master and the receiving Arduino alternates between slave and master modes.

Since we are using SPI to transfer values between the sensor unit Arduino and the receiving Arduino the transferred data consists only of updated sensor unit data. Therefore, we implemented an extremely simple SPI protocol: each transaction is 32 bits, with the first 16 bits corresponding to a unique identifier, which maps to a specific variable to update. The next 16-bits contain the update value of that variable. This allows us to transfer realtime data from the sensor unit to the receiving unit.

### (1) Sleep Quality Analysis

In order to try and compute a metric for sleep quality, we start by taking various statistics from the data we had available breathing rate, heart rate, and movement. We measure movement by taking raw accelerometer data and looking at the change in x, y, and z over many samples. We use this to calculate a value for each axis, and then add each axis's value together, resulting in a metric for overall movement. The algorithm looks for a few things to determine sleep quality. As a general rule, it looks for smooth, cyclic transitions in the heartbeat and breathing rate data.



For movement, we have observed that there can be a few spikes when rolling over or for other relatively small movements. We also had to add some detection for connection loss or sensor malfunction.

Occasionally, the nRF24L01 chips can fail to send data for a few seconds at a time, which causes a loss of sensor data. At these times, the data that gets logged is a 0, which is ignored by the algorithm when calculating the various metrics. These sensor errors were causing confusing outputs for us for some time, until we realized that issues like the user's finger temporarily coming off the heartrate sensor and wireless communication cutting out were twisting our results.

In order to calculate a sleep quality percentage, we look at the rate of change of each of the three metrics. Good sleep is characterized by smooth cyclic transitions of both heart rate and breathing rate, with relatively low movement during deep sleep. To use the chattiness of these metrics in our calculation, we first have to find these values. Below is a code sample that pulls statistics from the raw heart rate data. Once we have this data, we can empirically map it to various ranges to determine the quality of sleep[5].

We have been careful to provide proper references to all material that we used during testing, and code included in the final design that was not entirely written or inspired by ourselves.

We do not claim that the sleep quality metrics generated by this product are by any means absolute, the actual quality of sleep a user may experience can vary based on many personal factors. We claim that this product provides a good baseline for beginning to understand the factors that may influence sleep in both positive and negative ways, and giving our users access to data that they might otherwise be unable to obtain.

#### REFERENCES:

1. Glenn j. Landry, john R.Best and Teresa liuambrose 'measuringsleep quality in older adults: a comparison using subjective and objectivemethods'
2. Natural patterns of sleep.
3. Huckster.io 'sleep quality monitor'
4. Sleep cycle:types and stages.
5. Harvard medical on sleep.