Wireless Sleep Monitor

UCI Samueli School of Engineering
University of California, Irvine

Introduction

Many Americans struggle with sleep disorders and even more struggle to sleep well or enough for their average workload. To observe sleep, in-lab and home based sleep studies record useful information about a patient during sleep. This is often an expensive and uncomfortable process.

Objective

Our project aims to build a more affordable wireless sleep monitor. This monitor will be based off of a subset of the equipment used in a Polysomnography, the primary in-lab sleep study. The device, strapped to the patient, will record EEG and ECG data. This data will be wirelessly streamed to a server where it will be recorded, analyzed and could be presented to a physician.

Our Team

David Sargent	CSE — Signal processing and MSP430 software.
Alexander “Sasha” Sidenko	CSE — Raspberry Pi and Python development.
Can Vu	EE — Front-end PCB layout.
Steven Lam	EE — Front-end board schematic capture.

Acknowledgements

Thanks To:

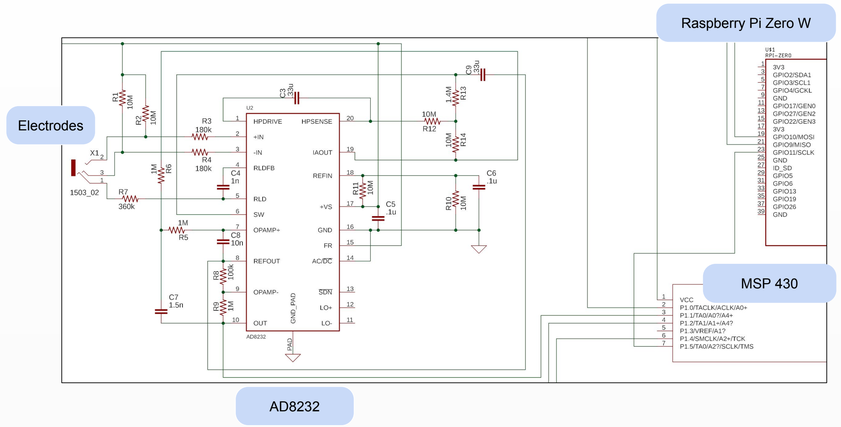
Our Advisor, Dr. Pooria Yaghini

Our Professor, Dr. Stuart Kleinfelder

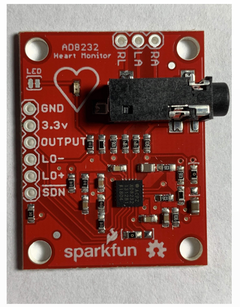
Our TAs, Beverly Angat Abadines

Arnav Vaibhav Malawade

Design



Signal Sampling



The first part of our design is a front-end board that contains the electronics needed to detect bioelectric signals useful for analyzing sleep.

- 3 EEG channels needed for detecting sleep stages.
- 1 ECG channel for monitoring heart rate.

Our design will use the AD8232 to amplify the bioelectric signals. This IC allows us to adjust the gain and analog filtering for each channel. The signal output is then sampled using a 16 bit Sigma Delta ADC on the MSP430F2013.

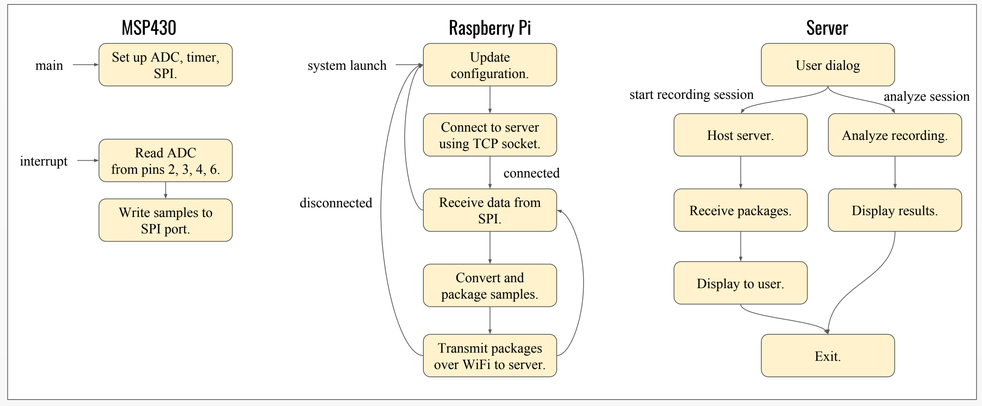
Our Progress: Designed an initial schematic in Eagle CAD (part of which is shown on the left).

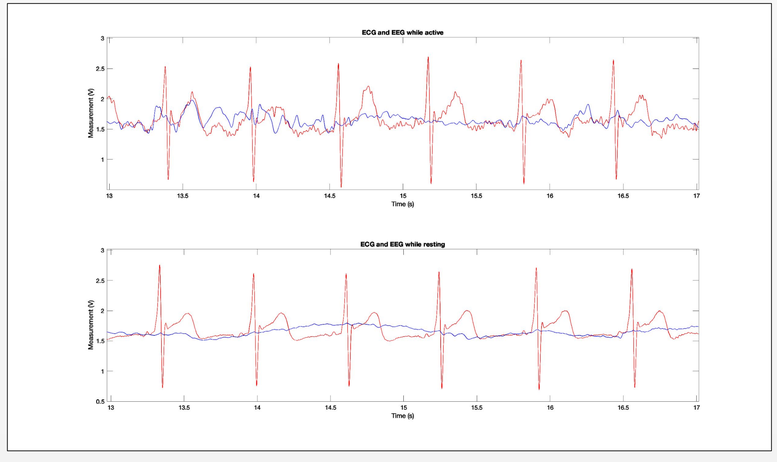
Data Transmission

The second part of design features a Raspberry Pi Zero W. The Raspberry Pi is in charge of various tasks including:

1. Powering the front-end board.
2. Using SPI to communicate with the MSP430F2013.
3. Connecting to a local WiFi network. Packaging signal samples and wirelessly transmitting data in real-time to a designated server.

Our Progress: Built Python modules to stream data generated by a Raspberry Pi to a host machine. Our overall software design is shown on the right.





Sleep Detection

The last part of our design will include a server that will process data from the Raspberry Pi. Its job is to:

1. Start/stop a recording session.
2. Connect to the Raspberry Pi Zero W over WiFi and write the incoming data to a file.
3. Present a graph of the EEG/ECG signals.
4. Give a summary of a recording session including the estimated time asleep and any detected anomalies.

Our Progress: Bought several SparkFun boards that include the AD8232 and powered them through a Raspberry Pi. Recorded test ECG and EEG data as shown on the left.