HappyLife

IDENTIFY AND COUNTER STRESS IN YOUR DAILY LIFE

Martynas Rimkevičius, Leonardo Vaninni, Baher Wahbe
Pervasive Computing | 2020-12-20

Table of Contents

Introduction

At the time of writing, the world has been engulfed in a life-altering global pandemic, and somehow this is not even the most stressful aspect of our lives. Everything from shopping, socializing, learning, and working have all in some way or another been transformed. These changes have had an enormous impact particularly on the mental well-being of many individuals, as most people are now unable to engage in most standard recreational activities such as meeting friends, seeing family, or playing sports, while simultaneously increasing their time on social media. This is all accompanied by the persistent worry of contracting the virus or the high likelihood of losing your job. Thus, it is more than an understatement to say that the average citizen of the world is presently under a substantial amount of stress.

We found this environment to be the perfect birthing place for our project 'HappyLife'. With its main goal being to help the user improve their mental health by allowing them to monitor, track, and analyze their stress levels by measuring their skin conductance using the signal produced by an epidermal activity sensor. With this project we want to grant the user the ability to visually see their stress, and even pinpoint the occurrence of a stressful event down to the minute. With the hopeful desire, that this allows the previously discussed average citizen to find out exactly when and what activities are inducing stress and monitor the effect of different stress coping mechanisms such as breathing techniques.

Literature study

To create this project, we had to first understand the human biological signals, we did this with the help of some literature.

Firstly, we wanted to find out what kind of signals do epidermal activity (EDA) sensors produce, what they represent and what we could gain from the data. This meant that we would need a source which described the biological side of sweat glands as well as technological – how to measure it, how to process the data. This information was described in an article "Innovations in Electrodermal Activity Data Collection and Signal Processing: A Systematic Review" [1] by Hugo F. Posada-Quintero and Ki H. Chon. Epidermal activity consists of skin conductance responses which represent an increase of sweat produced by sweat glands on the persons fingers. Sweat glands are used in a human's body for cooling but also are connected to the sympathetic nervous system. Skin conductance responses can be split into two categories: event-related experiments (strong or startle-like stimuli) or tonic stimuli response (such as stress, anxiety, arousal). These responses differ by latency and the reaction length to the event. While event related responses last for a few seconds, the tonic response might last up to tens of minutes after the event. To understand how to identify the event, the response had to firstly be found, as shown in the example in the article:

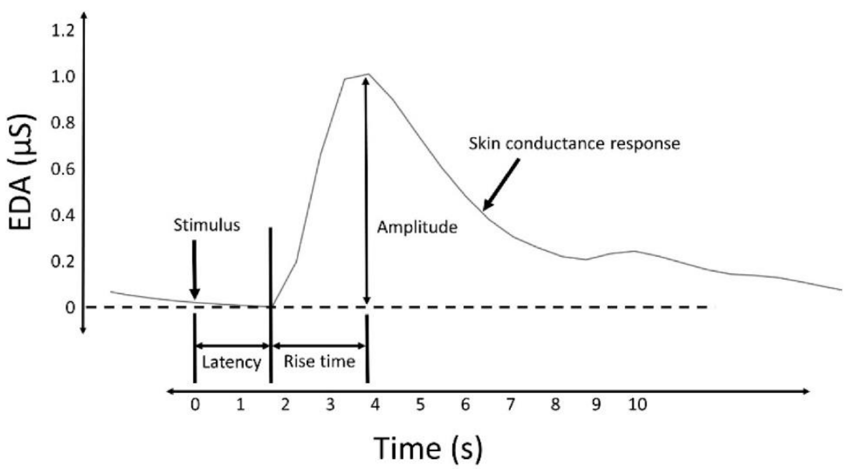


Fig. 1 EDA (measured in microsiemens) has an increase of amplitude after stimulus

Secondly, to find out how to process the data and get more information on conventional identification algorithms we read an article "What's your current stress level? Detection of stress patterns from GSR sensor data" by Jorn Bakker, Mykola Pechenizkiy, Natalia Sidorova [2]. This article describes a similar system to what we were trying to achieve – stressful event identification and reduction system, therefore the understanding of sympathetic system and techniques used to preprocess the data in their study were useful to us too. We found that the human neural system is constantly going through a cycle of normal state, aroused state, stressed state, and relaxing state. Having found this sequence, separate parts of it could be identified such as the ending of the "normal" state, where the event might have happened. This article also showed that there are several types of noise to account for: electrodes might not have the best contact with the skin because they are not worn tight enough, the device might be accidentally touched causing random and meaningless peaks in the data, and when the electrodes are applied to the skin, the skin sweat glands need a resettling period of around 15 minutes to reach a temperature equilibrium to show accurate data. Preprocessing techniques to remove various types of noise described in the article were: median filter (to filter out the noise caused by electronic devices surrounding the sensor and electrode connection loss, and to preserve the true form of the data), aggregate filter (to remove unnecessary peaks and lows, which would help identify tonic stimulus but not short-term stimulus) and data discretization filter (it was later used to identify stress in their system).

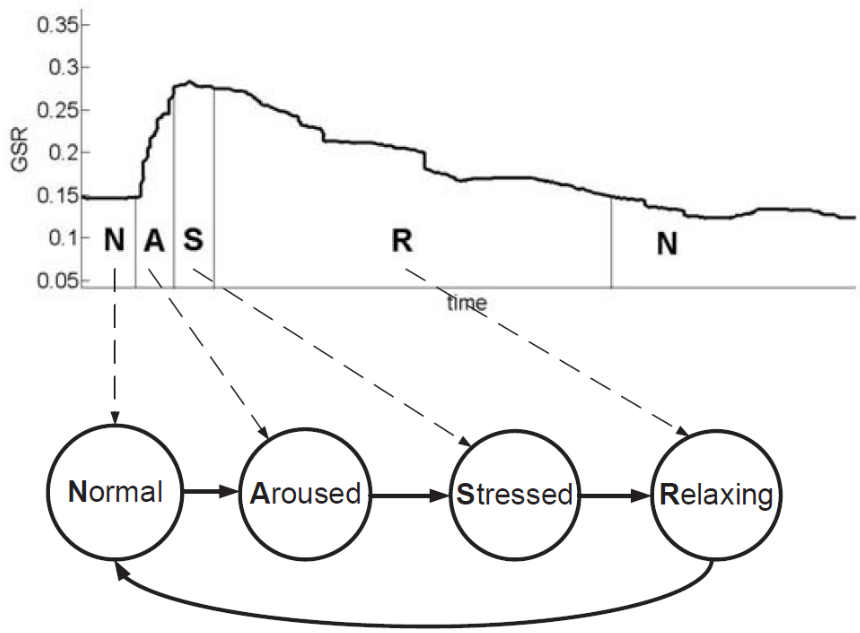


Fig. 2 Example of stress pattern mapping on the GSR data

Overall, we learned how human bodies show that the nervous system activity has increased and how to understand the data we are receiving from the sensor. Although there were some identification algorithms mentioned in both articles, we have focused our attention on understanding the neural system as well as what could cause the data to be corrupted and how to fix it, while having a challenge of producing the identification algorithm ourselves.

Concept of Operations

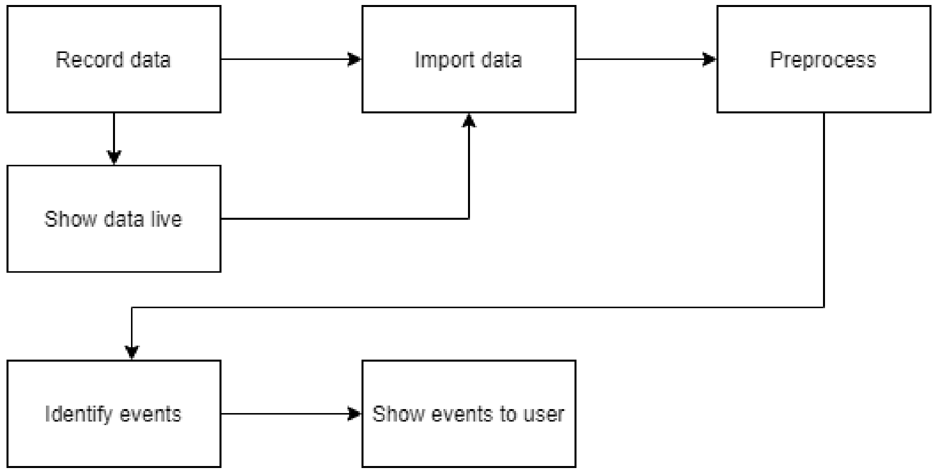


Fig. 3 block diagram of the system

Our system is made for the people who want to be aware of and improve their mental health. By creating this system, our main goal is to help people identify activities which cause a rise in their sympathetic system's activity. System's primary stakeholders could be identified as: people who want to reduce the amount of stress during their day, doctors who work with mental health and companies who want to improve their employees' mental state.

To achieve our goal of identifying stress-causing events we need the user to use this device for extended periods of time (minimum of 2 hours) because the sweat glands that are covered by the electrode need to adjust to the heat before producing reliable information on the neural system's activity and because the identifier is designed to spot long-term stress. The same case could be made for sports and environment, in which the temperature is changing. If the user were to do sports, they would have to rest for at least 20 minutes before using this system because otherwise the information would get distorted. Also, if the surrounding environment were to heat up while using the system, the human body temperature regulation would start making more sweat, thus creating false data unrelated to the neural system. Moreover, this system is collecting the user's personal data, thus there must be good security measures in place so that no information is leaked. With the current design, recorded data is stored on the user's computer and does not use any cloud services which could be broken into.

Our objectives while creating this system were oriented towards safety and ease of use. We aim to build a system which would not accidentally shock the user in any way, could be worn without discomfort, and results could be interpreted with ease. With the current prototype swift movements are not possible as the wiring is still flimsy. Though the electrodes are made such that they cannot shock and have a quick release function if anything were to happen. Also, our identification system should not confuse the user by spotting the places where there was no real event happening. This could lead to the user overanalyzing their schedule and making rushed decisions.

User could operate this system while at home or even while they are out of their stationary environments. It must be mentioned that the data may be distorted during high intensity physical activity, but it could work while walking, cooking, and interacting with people. With the current prototype it is not possible to go out and it could only be used in a stationary environment. One example user scenario with the current prototype could be this:

> User puts on the electrodes on the index and middle fingers and connects the microcontroller to their PC. Then they open the 'HappyLife' application and press the button 'Measure Now'. Now, when the system started to log data, they can do their own activities without moving too much out of their place as not to accidentally disconnect the electrodes. They can also start the live plotting function. By pressing 'Live Plotting' button they can start seeing live data from their sympathetic system.
>
> After some time, when enough data is recorded, user can disconnect the microcontroller from their PC, thus, saving the data in the folder 'Stress_Data'. If user wants to see the events which caused stress to them during that recording, they can press the button 'Import Data' and choose the recording they want to see the analysis of. After choosing one recording, they will see the events highlighted on the graph of the data and using the timeline of the plot they can cross-compare those marked events with their calendar, browser history or video call history to find out exactly what events were stressful to them throughout the measuring period.

In the future we imagine our system to be used throughout the day and made in such way that the user would not notice it while wearing or using it. Electrodes could be made to have wireless connection to the sensor and the microcontroller could be packaged in a neat way such that it would fit into a pocket or could be strapped to an arm. With these improvements user could use the sensor anywhere doing any type of activity and in the evening, they would connect the controller to their PC or mobile application and see the results. Also, the system would have full compatibility with calendars and browser history such that the user would not need to search for the events themselves, but the system would show events with names and timestamps in the day's overview. This inclusion, in turn, would help to exclude physical exercising, which distorts the data. Also, it would allow us to add additional functionality which would suggest to the user how to cope with daily stress. Overall, the system is intended to be used daily and for prolonged periods, thus allowing user to have the clear view of their day and ensures the application helps them as much as possible.

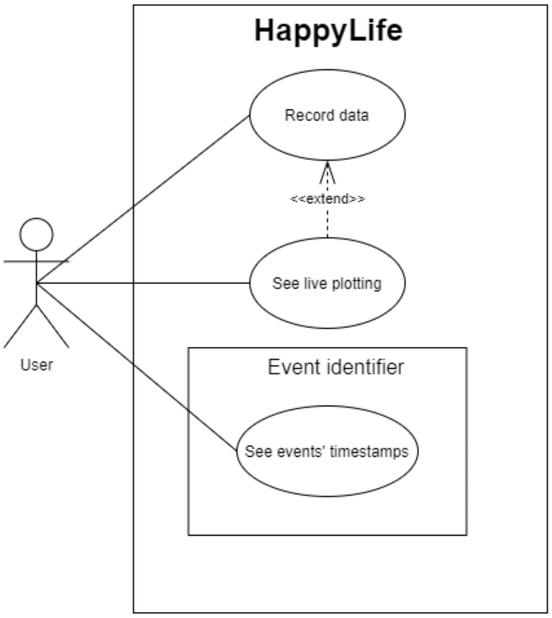
System Requirements

Our vision for the project was clear from the start, we knew what we wanted to do, and we knew more or less how to do it. Our only uncertainty was time, we did not know if we had enough time to fulfill all our requirements. It is for this reason that we chose the 'MoSCoW' method, allowing us to approach the implementation of our requirements hierarchically.

REQUIREMENTS

Requirement	Functional	Non-Functional
Must Have	– System must be able to identify events which cause stress to the user – System must show the identified events from user imported data	– System must identify the events in 3 seconds – System must identify the events with at least 50% accuracy – Recorded user data must be kept safe
Should Have	– System shall allow the user to monitor their stress levels with a live plotter functionality	– Design shall be clear and usable – While live plotter is enabled, the live plotter must refresh every 5 seconds
Could Have	– System could cross-compare the data analysis with user's schedule	– System could be carried by the user without disturbing them or being damaged
Won't Have	– System shall identify distinct types of stress	– Wireless connection between electrodes and sensor

Fig. 4 requirements ('MoSCoW' approach)



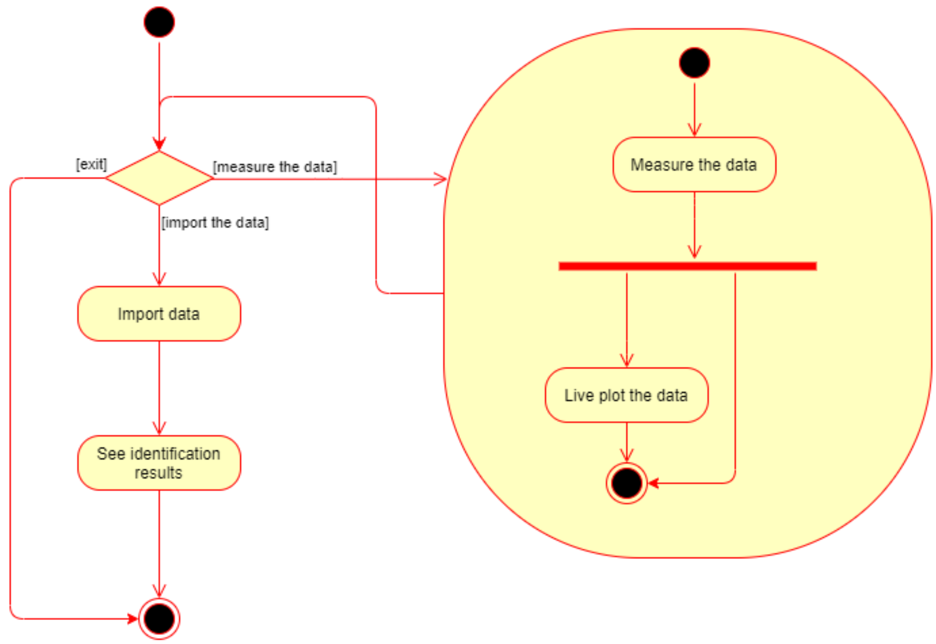


Fig. 5 Use case diagram of the system

Fig. 6 Activity diagram of the system

Upon launching the application, the user has the option to connect themselves into the system and start measuring their stress levels, this is done with a simple button click. After the measuring commences, an option to display their live stress levels is available, the user can choose whether they would like to or not. Ending the measurement phase is done by simply unplugging the device from your PC. This causes your data to be saved in an easily accessible file in your documents.

The user can also click on the import data button, this allows the user to import any of their previous stress data, running the identification algorithm on their data and displaying the results on the screen. The user can then check the magnitude and timestamps of various stressful events that occurred throughout their measuring session.

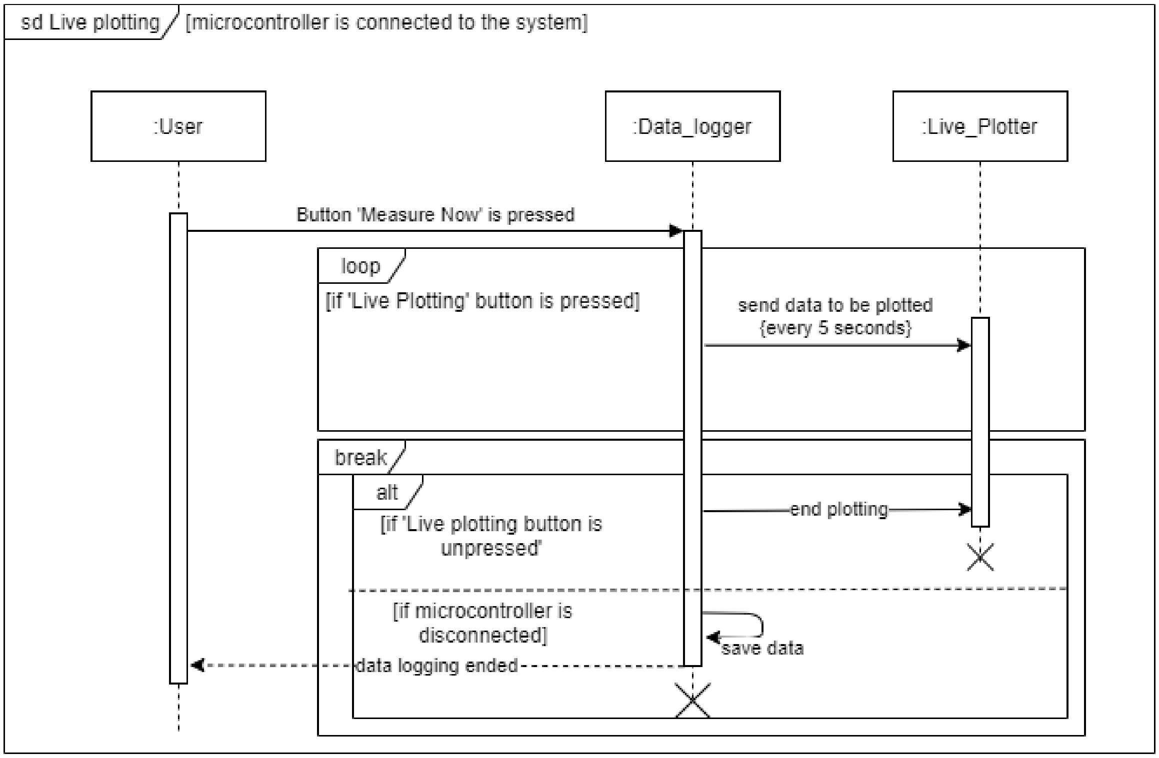


Fig. 7 Sequence diagram of Live Plotting functionality

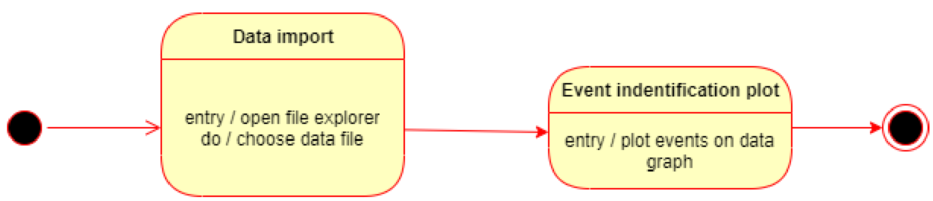


Fig. 8 State transition diagram for Event Identification functionality

System Design

COMPONENT LIST

- Sensor: MIKROE-2860
- Microcontroller: LPC1768
- Breadboard: AD-13 Advanced Solderless Breadboard
- Standard Electrodes and Electrode Cables

INTERFACING HUMAN SKIN

In order to measure GSR signals we had to interface a GSR sensor to a computer. The sensor we used is the MIKROE-2860. We chose this sensor because it was reported to be stable enough for accurate measurements.

The microcontroller we used is a LPC1768, connected to the analog interface of the sensor. The sensor has both an SPI and a 5V analog interface. Since the microcontroller's interface is rated at 3.3V we had to shift the signal with resistors.

The microcontroller was then connected to a computer using a USB cable.

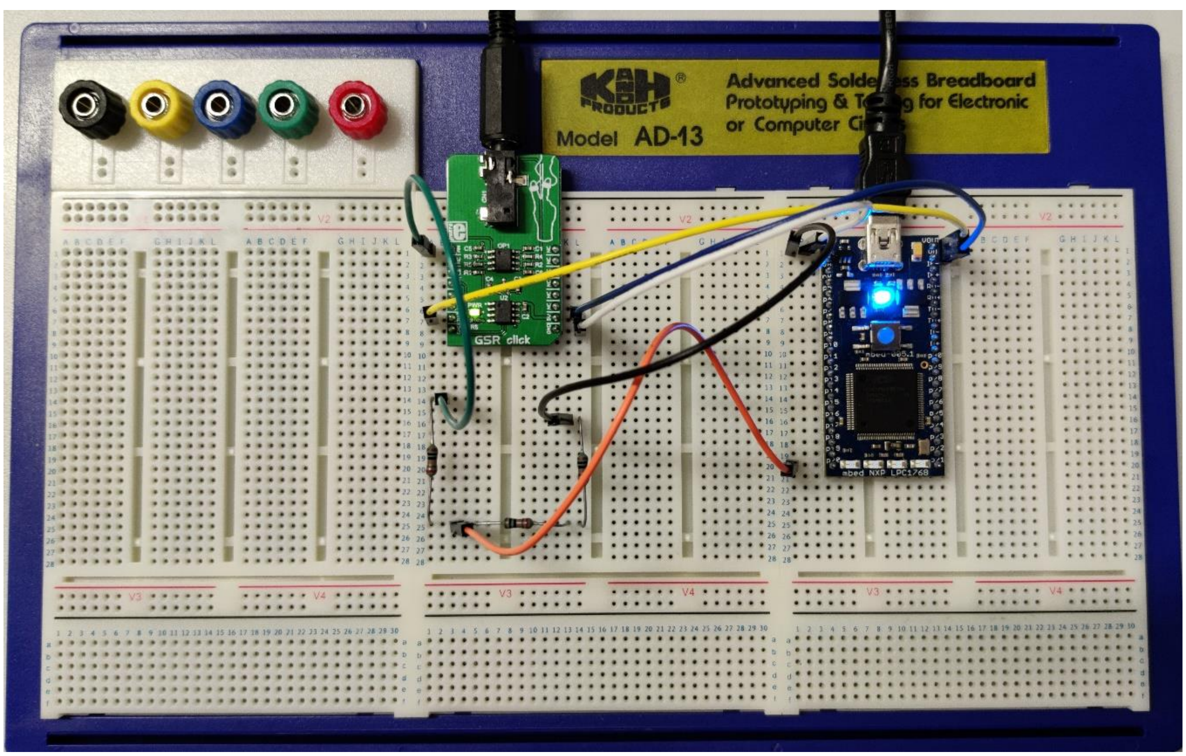


Fig. 9 system layout

The sensor interfaces the skin through a pair of generic electrodes put on the index and middle finger with conductive paste. The recommended type of electrodes are silver chloride electrodes without paste, but we could not get those. Still the generic ones, used in the prototype, turned out to be very consistent.

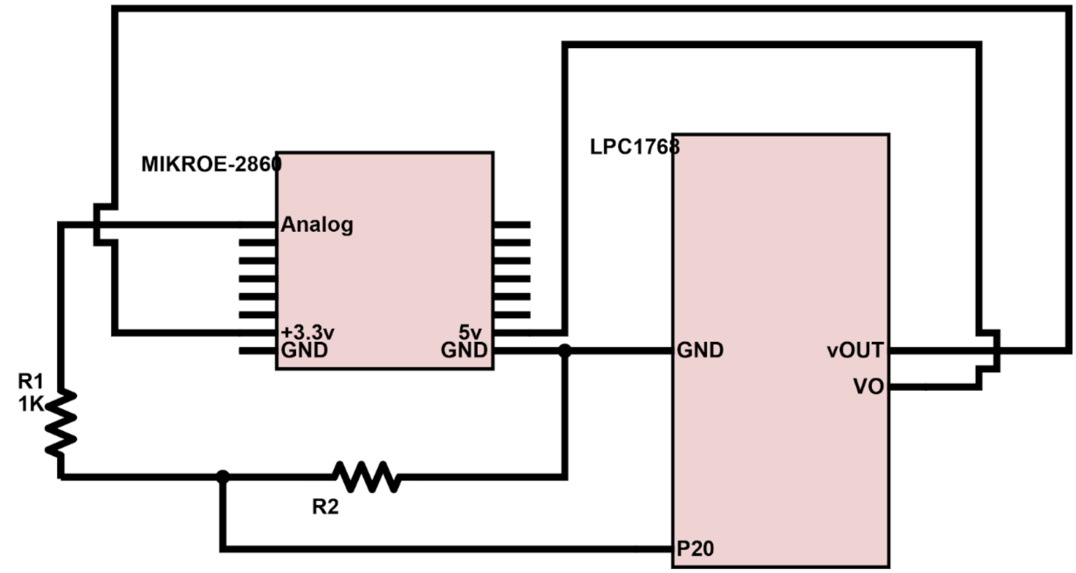


Fig. 10 system design diagram

While we were still experimenting with the sensor, we noticed that the first 15 minutes of every acquisition were very inconsistent, so we found that the electrodes need to be worn for at least 15 minutes before the acquisition. This is because electrodes temperature first needs to stabilize, as temperature interferes both with sweating and conductance.

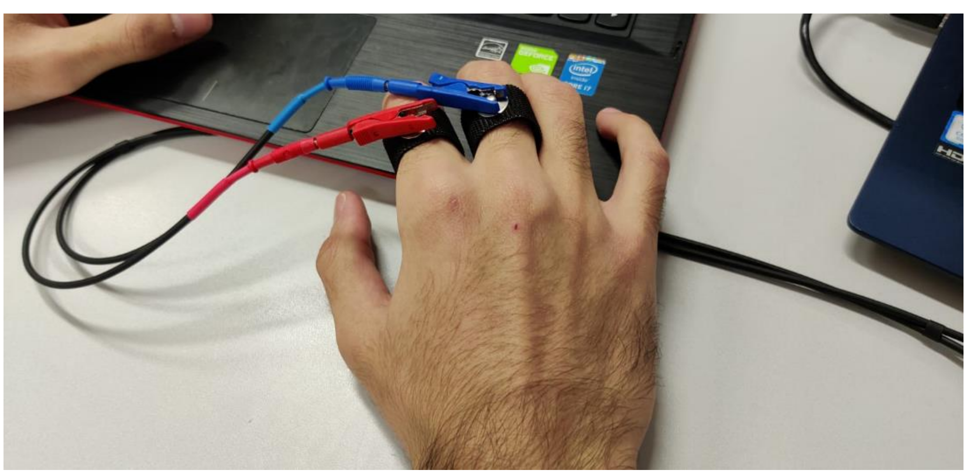


Fig. 11 interfacing human skin

LOGGER

To record data from the microcontroller we created a Python script that reads the serial port and saves in a CSV file. To make things easier to use we wrapped the logger in the GUI.

GUI

The GUI is written in MATLAB and allows to load and analyze a CSV file or to record a new one. It also includes a live plotter to visualize the raw data while it is being recoded.

EVENT IDENTIFICATION ALGORITHM

To identify the events in our system we firstly preprocess the data and then run the identification algorithm. Our preprocessing techniques consist of the median filter, which removes random noise in the data, and aggregation filter, which removes unnecessary peaks and lows in the graph. For identification we produced our own algorithm to detect the rise in the stress levels. Firstly, we search the data for positive slopes that rise at a constant rate. We check for positive gradients using the first derivative. We then look at the second derivative to see whether the slope grows at a constant rate, we do this by finding a high percentage of zero gradients in a particular timeframe. We are particularly interested in slopes that rise at a constant rate as this is a characteristic of the stress arousal period. This constant rising slope is, in turn, used to identify possible events that could have caused the stress. We find these events by searching for positive turning points that occur before our chosen slopes.

At this point our algorithm now has a list of possible stress inducing events, though some of them are incorrect, as our list might have more than one event per arousal slope while a stress inducing event and an arousal slope are mutually exclusive and have a one-to-one relationship.

To remove the redundant events, we iterate through our list, removing all events that lay on arousal slopes, keeping only the ones that are located before them.

On rare occasions our identification algorithm identifies two events which are extremely close together, to counteract this, we iterate over a final time removing all doubled events and combining them into one.

Ideally, for more accurate results, a neural network would have been preferred. We opted for our own identification algorithm as gathering tens of hours of training data for our neural network would have been impossible with our 3-week timeframe for the project (especially as our start was delayed while we waited for our components). Though, writing our own identification algorithm did have its advantages with it being much easier to test and much faster to run.

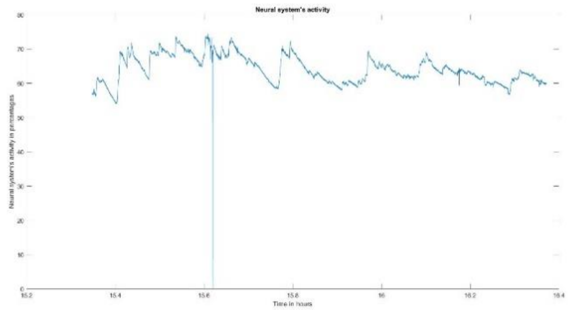


Fig. 12 raw data

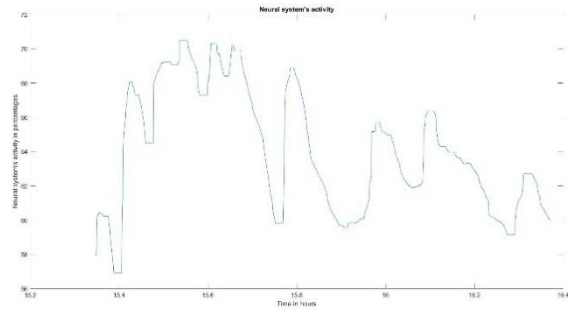


Fig. 13 data after median filter

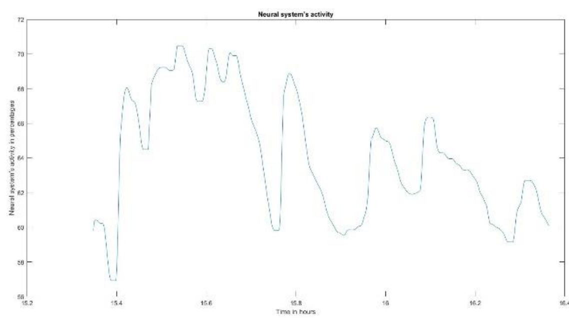


Fig. 14 data after aggregation filter

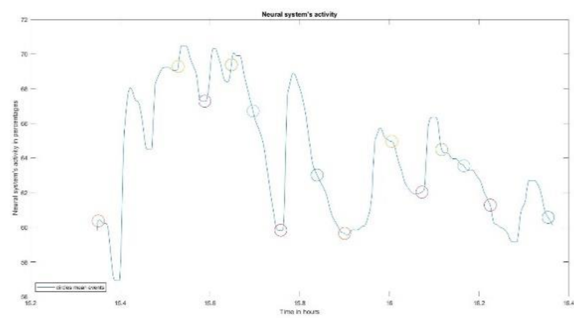


Fig. 15 data with identified events

Project Plan and Risk Analysis

MAIN WORK DISTRIBUTION

The project was split into 4 main parts:

- Building the system
- Gathering data
- Writing data analysis code
- Creating UI that combines system with code

This made splitting the work up between our team of 3 quite easy, each member of the team is assigned one main task, and a fourth task is worked on by the team.

Our criteria for task assignation were straightforward, each member chose the task they felt fit their skillset best. This ensured that the task was completed at the best possible standard the team could offer and that every members' motivation to work was high as they were working on something they like.

GANTT CHART

Tasks	Responsible Person	Progress	Start	Due	Start on Day*	Duration*
Research	Leo & Martynas	100%	10/23/2020	10/26/2020	0	3
Build The System	Leo	100%	10/26/2020	10/26/2020	3	0
Code The Data Logger	Leo	100%	10/26/2020	10/26/2020	3	0
Start Collecting Data From Sensor	Baher, Leo, Martynas	100%	10/26/2020	10/26/2020	3	0
Find Best Preprocessing Techniques	Martynas	100%	10/26/2020	10/26/2020	3	0
Find Best Event Identifying Strategy	Leo & Martynas	100%	10/26/2020	10/29/2020	3	3
Work On Feedback Strategy	Martynas	100%	10/26/2020	11/3/2020	3	8
Create Live Plotter	Baher	100%	11/6/2020	11/9/2020	14	3
Create GUI	Baher	100%	11/6/2020	11/9/2020	14	3

Fig. 16 Gantt chart

RISK ANALYSIS

Step	Risk Description
Conduct research and decide on project concept	– Project is uninteresting or unfeasible for our skillset and timeframe
Decide on our components	– Components are bad choices for our intents/purposes – Components are expensive
Build our system	– Incorrect connections causing circuits to fry
Write code to interact with sensor	– Python script out of sync with sensor microcontroller settings
Decide on preprocessing technique	– Our preprocessing technique loses too much data
Produce analysis algorithm	– Algorithm incorrectly identifies stressful events and confuses the user
Create feedback strategy	– Cluttered feedback prompts that confuse the user
Create the live plotter	– Too resource demanding and does not run smoothly
Create the GUI	

Fig. 17 risk analysis

Testing

Testing our system relied heavily on our human input as unlike our algorithm we understand the context, properties, and characteristics of an epidermal activity graph (described in the literature section fig 1), making it relatively easy to spot/identify stressful events in any dataset with no prior knowledge about it. Hence, to check whether our identification algorithm was functioning correctly, all we had to do was cross compare what we as humans knew were the correct answers versus what the system had come up with.

This means that the 'true' events happened where the system correctly identified the time or location of the real-world event according to us. However, during our testing, there were more events identified than needed and thus they were false if they were not in a correct position (before arousal period).

	True	False	Total
Event	18	12	30
Not an event	NaN	3	NaN

Fig. 18 confusion matrix

True negatives do not exist (not a number), as the data is continuous, and a true negative result is just not a stressful event.

It is important to note, that we are testing and evaluating a system that deals with biological signals, meaning we could never create the same data twice and have the same outcome.

Ultimately, our system had an 86% accuracy in identifying true positives (identifying 18 out of 21 events present in the data). Although, we had an abundance of false positives (40% of identified events were wrong), which might mislead the user. These thresholds are acceptable according to our initial requirements, though there is still much room for improvement.

Evaluation

Initially, as we pondered for a project idea, we were extremely interested in all the biometric signals our bodies could produce. This led us to contemplating how we could use these signals to improve certain aspects of mental health. We stumbled upon epidermal activity sensors and realized galvanic skin response sensors would be ideal for our use in identifying stress.

To say that we underestimated the complexity of the data we would be gathering is an understatement. Initially we had planned to spend only a day on research, we quickly learned a day was insufficient, causing us to add on an extra 2 days of pure research.

Our second obstacle was assembling the system, the sensor that we had chosen lacked proper documentation, leading us to a very confusing assembly process that took more time and effort than anticipated. This was an extremely important takeaway lesson, to never buy components without checking their documentation and ensuring its sufficient.

Thirdly, it was time to prepare the data acquisition phase. As a team, we are very delighted with the solution we came up with to access the sensor and log data from it. We used knowledge we acquired from outside the course, to write a sophisticated python script that was able to communicate with the microcontroller and store the data on the respective device. It reminded us to think outside of the box and use concepts learned outside the course in case we ever got stuck, which we did.

Recording data using our system was a challenging task, as we had only 2 lab sessions, which was not enough time given that our recordings ideally should be around 4 hours long. In retrospect, we should have been more insistent on our ability to take the system home, as it would have allowed us to increase our amount of recorded data significantly, potentially allowing us to create a, more accurate, neural network.

The analysis phase, at first glance, seemed easy. As humans we were confidently able to identify stressful events on an epidermal activity graph, and we thought it would be easily translated into code. But our lack of proficiency with MATLAB (compared to other programming languages we know) proved the task to be more difficult than initially anticipated. We struggled to differentiate casual rises in stress levels, from the actual stress arousal phase. We solved this, by locking ourselves in Leo and Baher's apartment until we found the solution. We ended up finding the solution by applying mathematical principles which we felt much more comfortable using. Lessons learned from this are: persistence will get us a long way when we run into a problem, and as mentioned previously, when you get stuck, don't be afraid to go back to the basic as we did here.

It was then time to implement the live plotter. We had expected it to be built quickly, but we learned about a python limitation that made us realize we needed more time. Our plan was to plot the csv file which the data was being written to every 5 seconds. But python locks all files it is writing to, hence we were unable to open it from MATLAB and plot it. We bypassed this by creating a second csv file that we updated every 5 seconds with new data, then locked, then plotted, in a cycle. This increased the time needed to build the live plotter and its complexity greatly. If we were to do this again, then we would choose a programing language closer to the hardware like C/C++, to avoid such built-in limitations.

Finally, to tie the whole project together, the GUI was built. Unlike most other processes, this one we expected to take an awfully long time but ended up completing in just one session. We credit this to MATLAB's app designer tool, which was very intuitive and simple to use.

Overall, we found these last 3 weeks of creating the project greatly enjoyable. It was a great introduction to system development, and we even got a taste of some agile development processes, as our schedules and plans changed on an almost daily bases. A lot went wrong that we learned from, and a lot went right that we can be proud of. We are very happy with the system we have produced.

References

[1] Posada-Quintero, H.F.; Chon, K.H. Innovations in Electrodermal Activity Data Collection and Signal Processing: A Systematic Review. Sensors 2020, 20, 479, doi:10.3390/s20020479

[2] J. Bakker, M. Pechenizkiy and N. Sidorova, "What's Your Current Stress Level? ("Stress Analytics in Education - win.tue.nl") Detection of Stress Patterns from GSR Sensor Data," 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, BC, 2011, pp. 573-580, doi:10.1109/ICDMW.2011.178.