IOT pulse measuring system

Project: IOT pulse measuring system
Deltager : Tobias Valbjørn
Projekt startdato: 27-aug-2018
Projekt slutdato: 14-dec-2018

Introduction

This report covers the 5th semester IOT project at EDE Herning in the fall by Tobias Valbjørn. It describes the development of a prototype of an IOT pulse measuring device, that can give you a general idea of your fitness level based on your resting heart rate. The project scope and requirements are described in Aarhus University course catalog for the course E5IOT.

Problem statement

The resting heart rate is the speed of the heart beat, when you are rested.

“As you get fitter, your resting heart rate should decrease. This is due to the heart getting more efficient at pumping blood around the body, so at rest more blood can be pumped around with each beat, therefore less beats per minute are needed. “[1]

It could be interesting to track the resting heart rate over a period of time, to get a general indication of fitness.

“The measurement of resting heart rate or pulse rate (the number of heart beats per minute) should be taken after a few minutes upon waking whilst still lying in bed. Give your body some time to adjust to the change from sleeping before taking your pulse (2-5 minutes). “

The measurement of the pulse first thing in the morning can be a challenge if you don’t have a wrist heart rate monitor, which can be expensive. If you use a chest strap, your resting heart rate will increase as you put it on. You can also use your fingers and a stopwatch, which can be difficult and inaccurate. In most of the cases you will have to remember the resting heart rate and note it manually.

Project description

The project focuses on improving and automating the process of keeping track of the resting heart rate.

The idea is that upon waking up, you would turn on your simple IOT resting heart rate monitoring system consisting of a particle photon, simple arduino heart rate sensor, a status RGB LED, and a vibrator with tactile feedback. The whole system is connected to your home Wifi.

Within a minute, as your body adjusts to change from sleeping, the system will be ready. Using the heart rate monitor your resting heart rate would be monitored, and uploaded to a webservice. You would get tactile feedback about when the test is done, and the results are

[1] https://www.topendsports.com/testing/heart-rate-resting.htm

uploaded to the web. Throughout the proces you could also check the status LED for the system state.

You could afterward see what your average heart rate is, if it has been increasing or decreasing, and it would place you into a basic category according to your sex and age, based on figure 1.

Resting Heart Rate for MEN

Age	18-25	26-35	36-45	46-55	56-65	65+
Athlete	49-55	49-54	50-56	50-57	51-56	50-55
Excellent	56-61	55-61	57-62	58-63	57-61	56-61
Good	62-65	62-65	63-66	64-67	62-67	62-65
Above Average	66-69	66-70	67-70	68-71	68-71	66-69
Average	70-73	71-74	71-75	72-76	72-75	70-73
Below Average	74-81	75-81	76-82	77-83	76-81	74-79
Poor	82+	82+	83+	84+	82+	80+

figure 1. General classification of fitness based on resting heart rate.

The webinterface can be used any time to change the information or watch simple charts of the progress. The system could be used to evaluate a fitness routine. The webservice is responsible for accepting basic user information, storing the information, being available for the IOT system throughout the day, and displaying the results, ie as shown in figure 2.

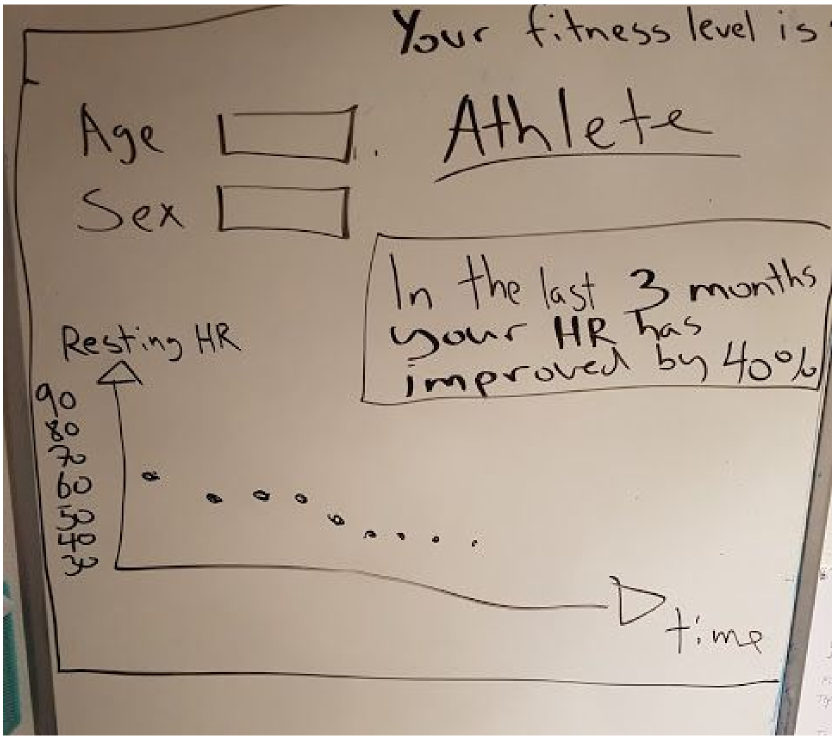


Figure 2. Demo webinterface.

Design requirements

The system will be turned on during the measurements. When the results have been read, the system will be turned off immediately. In this project scope to make a simple prototype and keep the budget low, USB power from the wall socket can be used instead of a battery powered system. Most people have a socket close to the bed for charging the phone or a night lamp for reading.
It does not need to take any sleep-mode functionality into account.
It only needs to start up fast.
It needs to be small to be in the bedroom, near the bed, supposedly at a bedside table.
The hardware needs to be organised in a way, that it can be easily equipped upon waking up.

Functional requirements

Need to have:

1. The platform has Wifi connectivity, and the system will connect over Wifi.
2. When the Wifi has been set up, the system should automatically connect to the internet when power is turned on.
3. After power is on, the application should start running automatically.
4. The platform has available digital or analog I/O to connect sensors and actuators.
5. The system reads the resting heart rate from a heart rate sensor local to the device.
6. The system should be able to send the heart rate locally from the particle photon to the webservice.
7. The device controls actuators in the form of LED and vibrator.
8. The device uses data from the webservice to augment what it does, by giving the user feedback on the LED and vibrator.
9. The webservice should be able to show the results of the heart rate measurements visually on a website.
10. The webservice should be able to store each measurement.

Nice to have:

11. The webservice should be able to compare the heart rate data with the heart rate table.
12. The webservice should be able to compare the heart rate data with previous data.
13. The system should be able to measure the heart rate accurately. Within 2 bps difference from a polar chest strap heart rate monitor.
14. The system should be able to store the necessary information of 1 user from the website. (age, sex, previous heart rate measurements, category, and date)
15. The whole system should be able to fit into a box with the maximum dimensions of length 15 cm, width 10 cm, height 8 cm.
16. The system should not take more than 30 seconds to initialize upon power on.
17. After the system is ready, the user should be able to start measuring the heart rate within 15 seconds.
18. The device will be able to connect to AU's “AU Gadget network”

Non-functional requirements

- User interaction will be kept on a minimum to keep the system simple.
- The system functionality will be focused on the task of measuring the resting heart rate.
- Existing standards and protocols will be used wherever possible.
- The system needs to be simple, with as few user interactions as possible.
- The system will be calm when everything is working fine. It will not bring attention to itself beside when giving the user instruction or feedback.

Hardware

The particle photon can be used, since the project has simple needs, and a small form factor would be prefered. There needs to be Wi-fi and a few GPIO's. We don't need a lot of GPIO's, memory, high processing speed, ports or other communication modules. Furthermore the device does not have to be able to deliver power to a lot of sensors and actuators, only one heart rate sensor, a RGB-LED, and a vibrator motor, where the particle photon will be sufficient. The particle photons tasks will be to measure an analog value from the heart rate, publish the analog value, get basic response from the internet, and control the LED and vibrator.

I can use the particle photon from the class set.

Interface analysis

Particle photon peripherals and GPIO

The particle photon has the following peripherals:

Peripheral Type	Qty	Input(I) / Output(O)	FT[1] / 3V3[2]
Digital	18	I/O	FT/3V3
Analog (ADC)	8	I	3V3
Analog (DAC)	2	O	3V3
SPI	2	I/O	3V3
I2S	1	I/O	3V3
I2C	1	I/O	FT
CAN	1	I/O	3V3[4]
USB	1	I/O	3V3
PWM	9[3]	O	3V3

The supply current limits are:

Parameter	Symbol	Min	Typ	Max	Unit
Supply Output Current	$I_{VIN-MAX-L}$			1	A
Supply Output Current	$I_{3V3-MAX-L}$			100	mA

If I use the Vin I can supply 1 A, and if I use 3V3 I can use 100 mA.

From the I/O characteristics these important characteristics are found:

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input/Output current max	I_{io}				±25	mA
Input/Output current total	$I_{io\ total}$				±120	mA

From one pin we can maximally draw 25 mA, and from all the pins it can maximally draw/deliver 120 mA.

General requirements for sensors and actuators

Any sensor or actuator cannot draw more than 25 mA from one I/O pin.
In total the current Input/Output from the pins cannot exceed 120 mA.

Sensors

We need our system to interface with a sensor.
The information we need from the outside world are a resting heart rate from a person converted to an electric signal.
The particle photon has 8 A/D inputs, and we can choose anyone of these:

A0~A7	12-bit Analog-to-Digital (A/D) inputs (0-4095), and also digital GPIOs. `A6` and `A7` are code convenience mappings, which means pins are not actually labeled as such but you may use code like `analogRead(A7)`. `A6` maps to the DAC pin and `A7` maps to the WKP pin. A4,A5,A7 may also be used as a PWM[2] output.

General information about sensors

There are mainly two kind of heart rate sensors that I am looking at. It is chest strap and optical sensors.

Chest strap

Chest strap heart-rate monitors use electrocardiography to record the electrical activity of the heart. A pad with electrodes records the electrical activity of the heart, but the pad needs to be moist for proper conduction of the electrical signal. It is both expensive and impractical since the system needs to be used when waking up. Applying moist on the sensor is not an option.

Optical sensors

Another technology is using an optical sensor, and this is the most common in wearable heart rate monitors. [2] They use photoplethysmography (PPG). It is the use of light to capture the flow of blood through the veins. Often they use LEDs with green light unto the skin. The different wavelength of the light will interact with the flowing blood, and the reflection of the light will be registered in a sensor. I is an integrated sensor, the received signal are amplified and noise cancellation is applied. The sensor output is an analog electronic signal. The optical sensors can be less reliable since they are placed further from the heart. It is an acceptable tradeoff, but the accuracy will be tested. Since there is amplification involved we will need an active sensor, which requires an external power supply.

Design requirements for heart rate sensor

It needs to be inexpensive, so it can be used in a low budget prototype system.
It needs to be compatible with the particle photon, and within the general requirements for sensors and actuators.
It needs to be well tested and well documented.
I am looking for one to put on a finger or wrist.
I need an optical heart rate sensor
It has to include amplification and noise cancelation
It has to be easy to apply.

Selection criterion

1. Accuracy
    - Difference between the measured value and the “true” value.
    - It has to be within 2 difference from a polar chest strap heart rate monitor.

2. Precision
    - The ability to reproduce with a given accuracy
    - Precision can be difficult to measure with resting heart rate, since subjective factors can change the outcome of the measurement. The heart rate could increase or decrease while measuring. Despite that the sensor needs to have precision. A counter measure could be to measure over time and observe the beats per minute over several minutes.
    - If we measure the heart rate over 5 minutes the bpm for each minute should not deviate more than 3 bpm.
3. Resolution
    - The smallest change the sensor can differentiate.
    - The sensor needs to be able to measure a difference of 1 bpm.
4. Response time
    - The time lag between the input and output
    - The sensor needs to receive the analog signal from the sensor immediately.

[2] https://arstechnica.com/gadgets/2017/04/how-wearable-heart-rate-monitors-work-and-which-is-best-for-you/

- I am working with electric signals, and I assume that this will not be a problem. The only test I will make of the response time is, if there is a visible lag of more than 1 second.

SEN-11574

There is a well documented plug and play heart rate sensor for arduino, SEN-11574[3], which means it is compatible with the particle photon. It has a lot of code to get started. It includes velcro, ear plugs, and other stuff to help with the measurements as seen on figure xx:

Figure xx SEN-11574 pulse sensor kit.

The maximum ratings has to be compatible:

Absolute Maximum Ratings	Min	Typ	Max	Unit
Operating Temperature Range	-40		+85	°C
Input Voltage Range	3		5.5	V
Output Voltage Range	0.3	Vdd/2	Vdd	V
Supply Current	3		4	mA

It can be used with 3.3 V and 5V, and it has a supply current of 3-4 mA which is well within the boundaries of our system.

[3] https://www.sparkfun.com/products/11574

It has 3 pins:

RED wire = +3V to +5V
BLACK wire = GND
PURPLE wire = Signal

It can be connected to the particle photon in the following way:

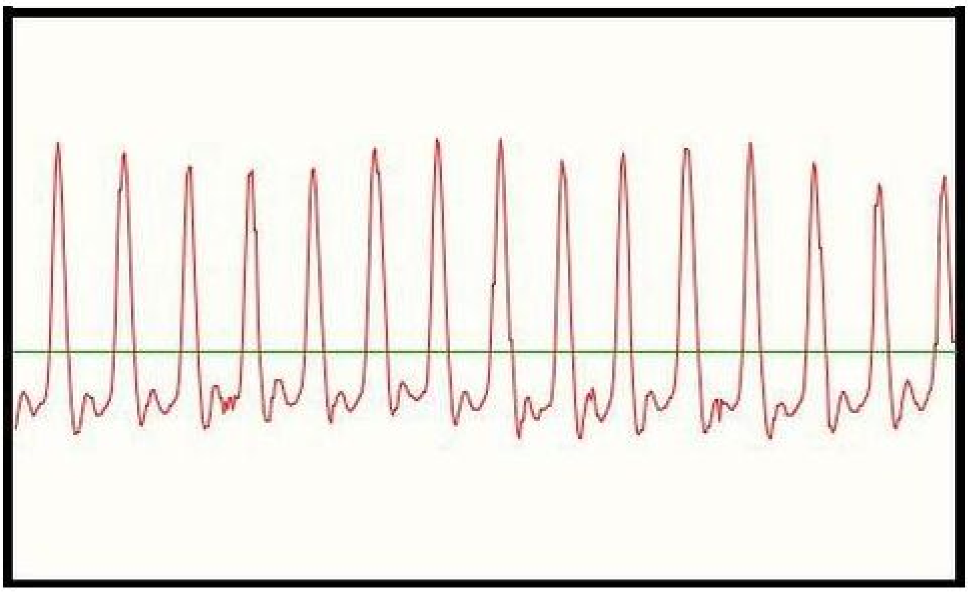
Sensor	Particle photon
Red wire	5V
Black wire	GND
Purple wire	A0

I use the 5V as it can deliver most current to the connected devices.

The price is: $24.95 = 161 dkk, which meets the requirement of low budget.

Sensor description[4]

The sensor uses an ambient light sensor from Avago and a reverse mount LED from Kingbright. It uses a “filter and amplifier to increase the amplitude of the pulse wave and normalize the signal around a reference point.” Without contact to a finger the analog output from the sensor will be around Vdd/2. When it is applied to a finger, the analog voltage will fluctuate around the reference point when the blood flow from the heartbeat reaches the finger as seen in figure x:



[4] https://pulsesensor.com/pages/open-hardware

Figure x. Example measurement showing bpm. The green line represents the reference at vdd/2.

The Arduino can be programmed to register, when the signal rises above the mid-point.

Documentation

I have been looking at the documentation for the sensor, and from a quick search it seems that the sensor is well used and described in hobbyist projects:

The sensor is made by “World Famous Electronics”, which actively maintains example projects and code at:
https://pulsesensor.com/

Getting started code:
https://github.com/WorldFamousElectronics/PulseSensorStarterProject

Getting advanced code:
https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino

Pulse sensor processing visualizer:
https://github.com/WorldFamousElectronics/PulseSensor_Amped_Processing_Visualizer

Project with code and thingsspeak integration:
https://circuitdigest.com/microcontroller-projects/iot-heartbeat-monitoring-using-arduino

Conclusion

The SEN-11574 meets the design requirements and are chosen for the project. Given the information I have about the sensor, I can only guess that the chosen sensor meets the selection criteria, it must be tested during the development of the prototype.

Actuators

Several types of have been considered. A LCD-display, a RGB LED and a vibrator motor.

LCD display

The LCD display was initially considered because of the possibility to show instruction and the output of the pulse measurement on the screen for the user to see immediately. In that way the system could be used locally without an active internet connection, which is in line with graceful degradation. The biggest counter argument was that it makes the system bigger and more complex, and it adds light and information that the user might not want first thing in the morning. The bigger system is less practical, and the added information could increase the heart rate. Another point is that the system’s main purpose is to measure the heart rate, the screen could be redundant, seeing that I want to show the information more visually pleasing on a website anyway. If the user wants information afterwards, he can just visit the website on his smartphone.

For the system to be in accordance with the project requirements, there needs to be actuators that acts on information from a web service. To keep it simple, the actuator will show the status of the process. I need an actuator that can inform the user of the status of the project. When is the system ready to measure pulse? When is the system measuring? When is the system done, and have received acknowledgement from the web server? An LED can do the job. It will not disturb the user, and it is lightweight. I considered using the onboard LED, but it was not sufficient to display all the different information.

RGB LED

The RGB LED can show the status throughout the whole process. In the end it will give a signal when the webservice has received the information. Figure xx shows a use case with the output of the diode. A RGB LED can display different colours, and therefore show the status more clearly. In E4BIS I used an RGB LED for a design, where I tested it in action. There are no special requirements for the RGB LED. Model No.: YSL-R596CR3G4B5C-C10. In figure xx. pictures of the LED are shown.




Figure xx. Pictures of the RGB LED. In different scenarios. First unconnected, then blue, green and red.

We can see from the datasheet that the forward current is 20 mA.

Pararmeter	Symbol	Red	Green	Blue	Unit
Forward current	I_F	20	20	20	mA

It is the amount of current that lets the LED shine the brightest.

I need to know the forward voltage of the LED to see how the diode is supposed to be powered.

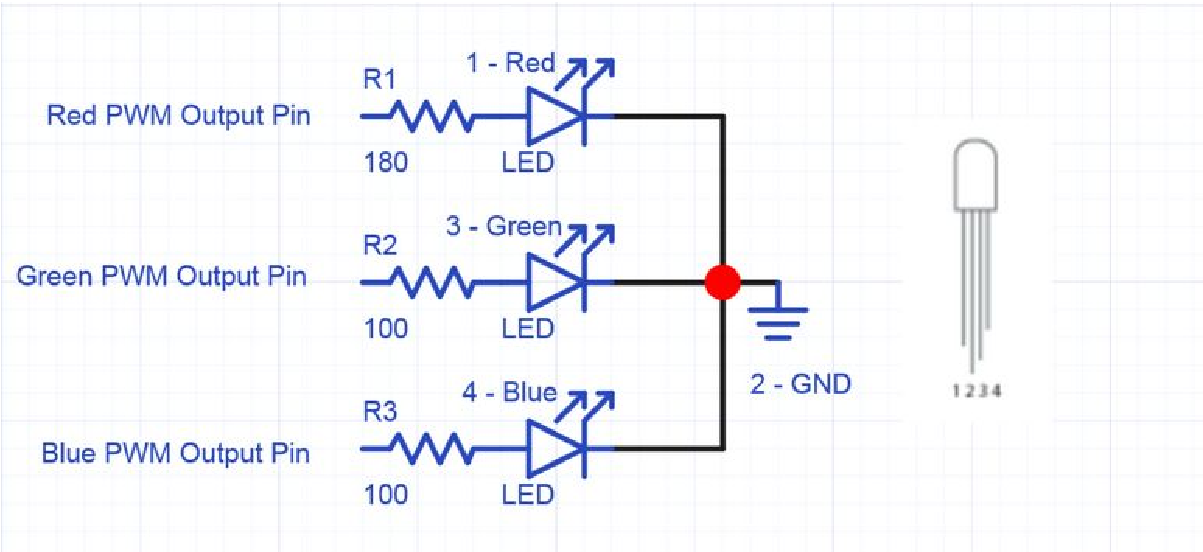
ITEMS	Color	Symbol	Condition	Min.	Typ.	Max.	Unit
Forward Voltage	Red	V_F	I_F=20mA	1.8	2.0	2.2	V
	Green			3.0	3.2	3.4	
	Blue			3.0	3.2	3.4	

I can see from the max forward voltage of 3.4 that 3V3 will not suffice. We need 5V to power this diode. I look in the datasheet to see if there is a pin that can supply that:

Pin	Description
VIN	This pin can be used as an input or output. As an input, supply 3.6 to 5.5VDC to power the Photon. When the Photon is powered via the USB port, this pin will output a voltage of approximately 4.8VDC due to a reverse polarity protection series Schottky diode between VUSB and VIN. When used as an output, the max load on VIN is 1A.

VIN when powered through USB can delieveer 4.8V. I will power the system using the USB, so I have 4.8V accessible to supply the diode, that will suffice. VIN can deliver up to 1A compared with 3V3 that can maximally deliver 100 mA. When the current for the LED will be around 16 mA, it is a plus, that we do not draw that current from the 3V3, seeing that it would use 16% of the total current available.

In E4BIS I used the following schematic:



Current delivered to red diode:

$$4.8V - 2\ V \div 180\,\Omega\ =\ 15.55\ mA$$

Current delivered to blue and green diode:
$4.8V - 3.2\ V \div 100\ \Omega\ =\ 16\ mA$

The RGB LED needs 3 PWM pins. The particle photon has 9 - RX, TX, A4, A5, A7, D0-D3. I will use D0-D2

RGB LED (Actuator)	Particle photon
1-Red	D0
3-Green	D1
4-Blue	D2
2- GND	GND

Vibrator motor

It is not certain that a user would like to keep an eye on the LED throughout the pulse measurement. He might want to lie in the bed with his eyes closed, while the system monitors the pulse. Haptic feedback could enable him to do that. With his eyes closed he could get feedback from a haptic sensor throughout the pulse measurement, as well as a signal when the measurement is done. A vibrator motor can be used to provide the haptic feedback. In a design in E4BIS I have worked with a Precision motor driver Model 308-100, where I used it to create signals indicating “ready”, “busy”, “alert” etcetera. It worked well for the purpose. The only requirements I have for the vibrator motor are that it is compatible with the system-2-be. If it is, I will reuse the one that I have already used. Figure xx shows a picture of the vibrator motor:

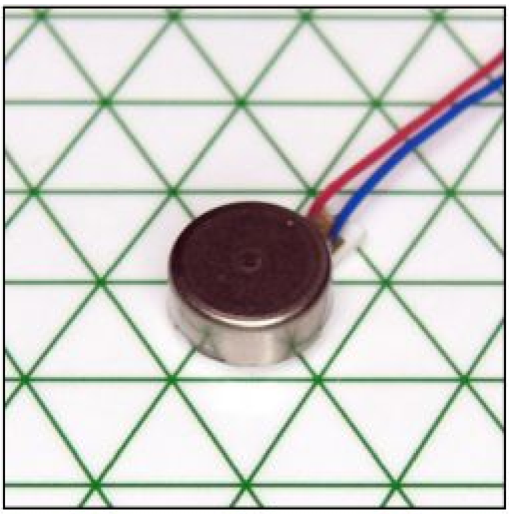


Figure xx. a picture of the vibration motor.

The typical operating current is:

Typical Rated Operating Current:	70 mA

And the operating voltage is:

PARAMETER	CONDITIONS	SPECIFICATION
Rated Operating Voltage		3 V
Max. Operating Voltage		3.3 V

The 3V3 pin is needed for operation of the vibrator, and it typically uses around 70 mA. The 3V3 can maximally deliver 100 mA, so the motor will use around 70% of the total current available. 70% is a lot, but every other sensor and actuator could operate on 5V, that can deliver up to 1A. The vibrator motor will be the only device using the 3V3, and the power consumption can be accepted.

The motor only has to terminals, a red and a blue. The polarity does not matter for motors. The motor will be vibrating heavily if I connect it to the 3V3 pin directly. I need to be able to regulate the strength of the vibration. I can use the same circuit i used in E4BIS, as seen in figure xx:

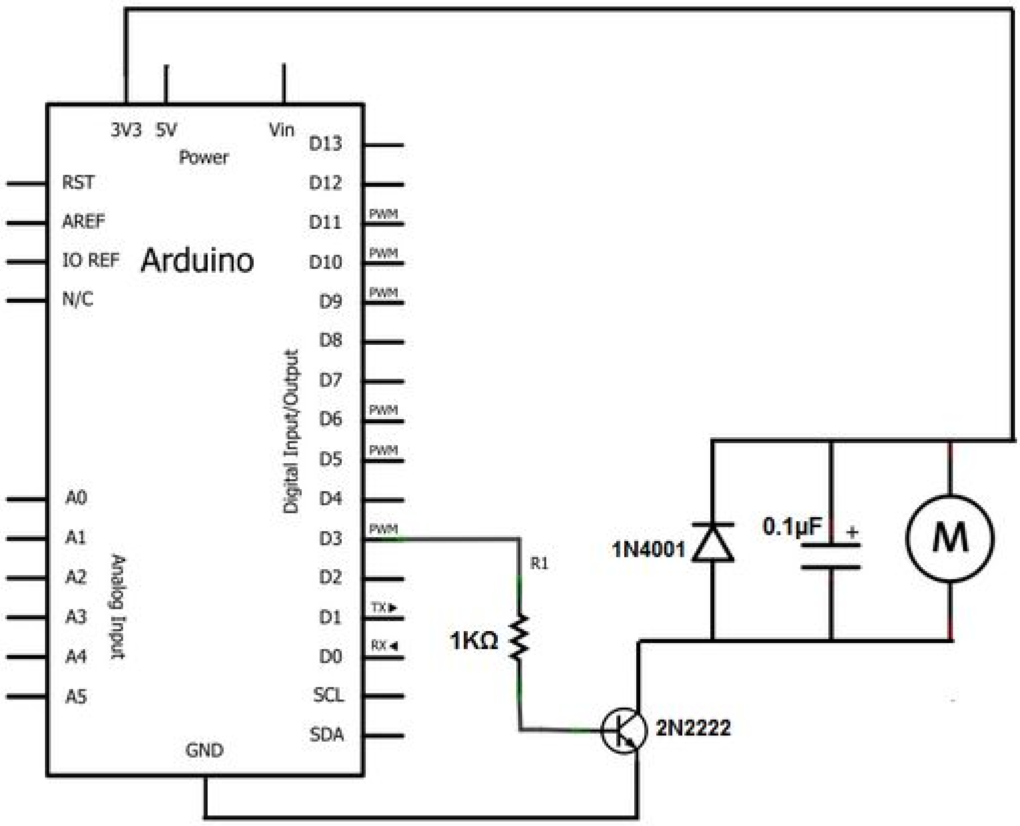


Figure xx. A schematic of the circuit used for the motor. In our case we don't use an arduino but the particle photon.

The PWM is used to switch on the transistor, that will regulate the strength of the vibrator motor. The PWM D3 can be used.

The circuit is from a website, where there is a great explanation of the workings of the circuit[5]:

“The diode acts as a surge protector against voltage spikes that the motor may produce. The windings of the motor notoriously produce voltage spikes as it rotates. Without the diode, these voltages could easily destroy your microcontroller, or motor controller IC or zap out a transistor.”

“The 0.1µF capacitor absorbs voltage spikes produced when the brushes, which are contacts connecting electric current to the motor windings, open and close.”

“To make sure that too much current does not flow from the output of the transistor, we place a 1KΩ in series with the base of the transistor. ”

[5] http://www.learningaboutelectronics.com/Articles/Vibration-motor-circuit.php

Pinout showing the connection between the vibrator motor and particle photon:

Vibrator motor (Actuator)	Particle photon
collector at transistor	D3
Red	3V3
Blue	GND

Overview

A complete table of the particle photon pins and the devices that are connected to them:

Particle Photon Pins	Device using
3V3	Vibrator motor
5V	Heart rate sensor, RGB LED
GND	All devices
A0	Heart rate sensor
D0	1-Red
D1	3-Green
D2	4-Blue
D3	Base terminal of transistor in vibrator motor circuit

A simple overview of the logic in the system:

Quantity being measured	Input Devices (sensors)	Output devices (Actuators)
Heart rate	Heart rate sensor	RGB LED Vibrator motor

Diagrams

Figure 3 shows the architecture of the system. Figure 4 shows a simple use case diagram, with the flow through the system. The diagrams can be open to changes as more information is gathered, especially figure 4, and the implementation of the webservice. A webhook might be used as middleman between the local system and the webservice, and maybe there will be more than one web interface to store the information and represent the results.

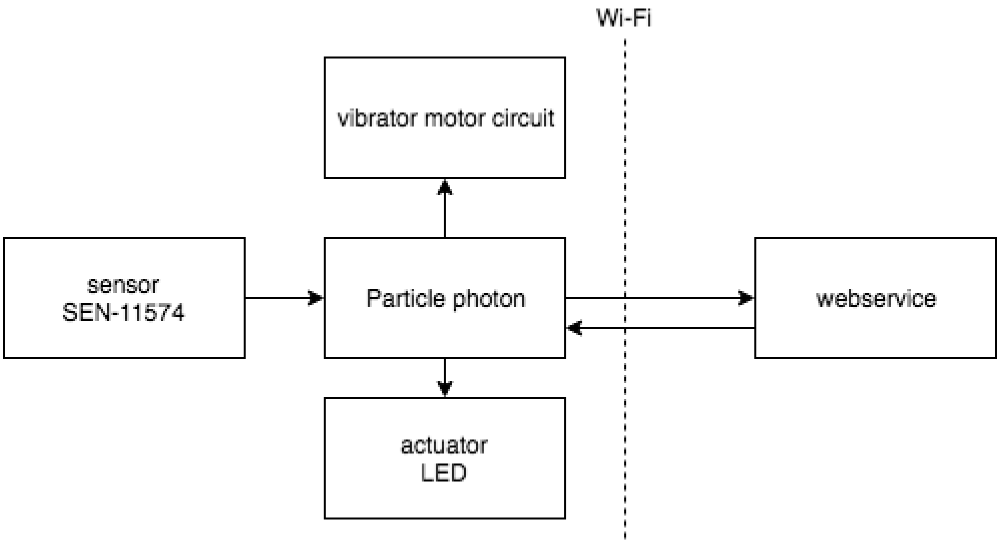


figure 3. architecture of the system.

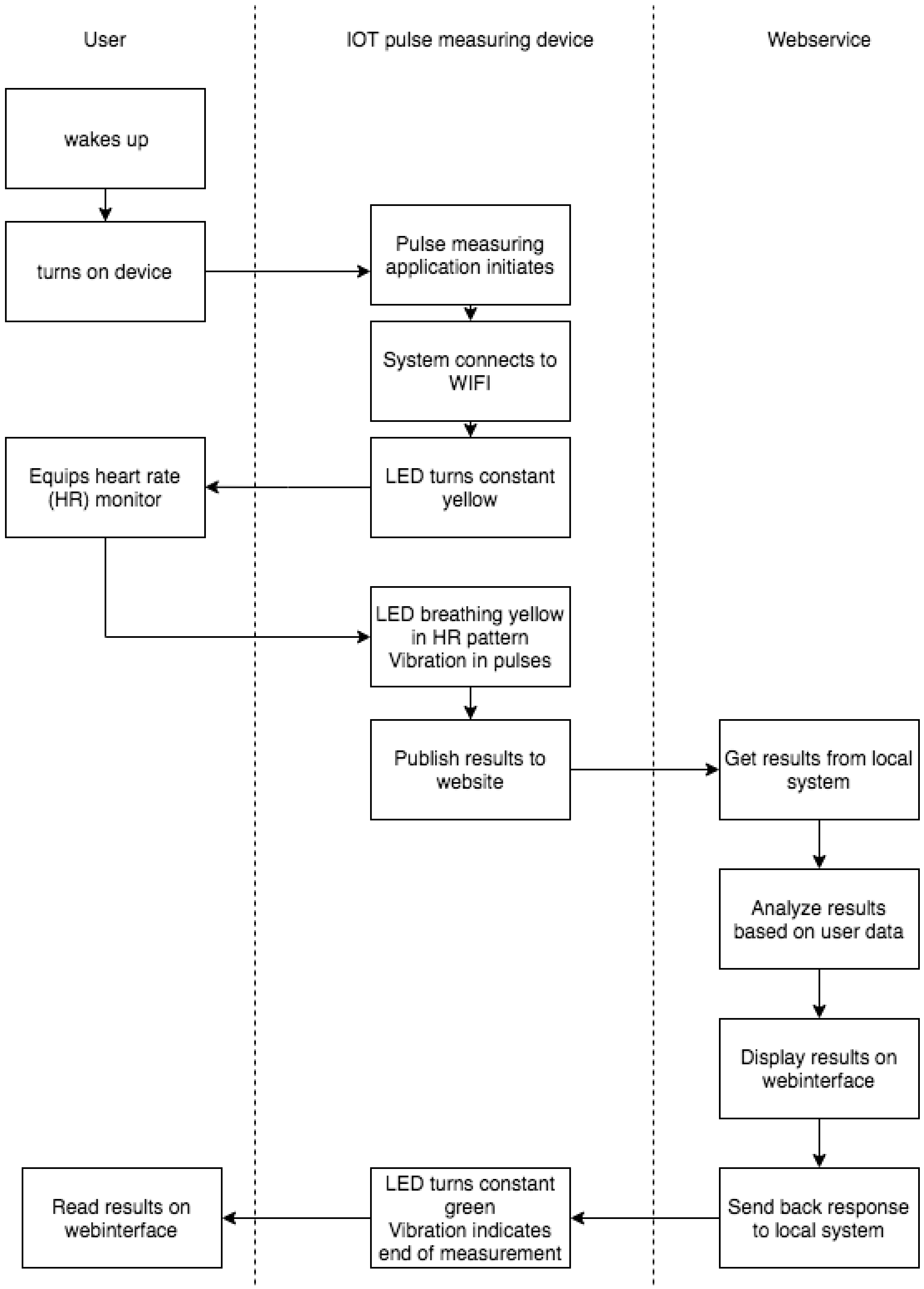


Figure 4. use case diagram for the system

Plan

Expected

Uge	Assignment	Deadline
35-37	Project selection Project description Purchasing hardware	14-sep
38	Implementation of heart rate sensor code Presentation	21-sep
39	Implementation of heart rate sensor code	28-sep
40	vibrator and RGB-LED implementation user experience testing.	5-oct
41	Get the whole heart rate measuring system working locally	12-oct
43	Analysis and design of the webservice part. Presentation	26-oct
44	Send local data to webserver Response from webserver Implement webservice to store and display resting heart rate.	2-nov
45	Implement analysis of user data	9-nov
46	Implement analysis of user data	16-nov
47	Represent user data	23-nov

	Presentation	
48	Implementation of input user data	30-nov
49	Final adjustments, report	7-dec
50	Report Upload via wiseflow	14-dec

Actual

Uge	Assignment	Deadline
35-37	Project selection Project description Purchasing hardware	14-sep
38	Project specification. Requirements and documentation with focus on interface analysis of sensors and actuators .	21-sep
39	Implementation of heart rate sensor code	28-sep
40	vibrator and RGB-LED implementation user experience testing.	5-oct
41	Get the whole heart rate measuring system working locally	12-oct
43	Analysis and design of the webservice part. Presentation	26-oct
44	Send local data to webserver Response from webserver Implement webservice to	2-nov

	store and display resting heart rate.	
45	Implement analysis of user data	9-nov
46	Implement analysis of user data	16-nov
47	Represent user data Presentation	23-nov
48	Implementation of input user data	30-nov
49	Final adjustments, report	7-dec
50	Report Upload via wiseflow	14-dec