**Lesson Plan Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject: | Biology | | | | |
| Authors: | Teachers: Mgr. Soňa Patočková  Students: Josef Juřička, Tereza Frantová, Šimon Mareček | | | | |
| Date: | | | | \_\_\_\_\_/\_\_\_\_\_/ \_2018\_ | |
| Estimated time: | | 50 minutes + 50 minutes |  | |  |
|  | |  |  | |  |
| Summary: | Connecting Biology and digital technologies. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objectives**  (Specify skills information that will be taught) | **Activity/ Information**  **Teacher Guide/ Student guide** | **Materials Needed**  (Other resources - web, book...) | **Assessment Methods**  (steps to check for student understanding) | **Time**  **Where?** |
| \* Work with Arduino and a sensor used to take pulse (connected with ICT and programming)  \* understanding how a human´s heart works during an exertion  ***Know Concepts or Keywords :***  - Arduino  - Heart sensor  - heart  - heart´s activity | **Motivation Activities**  How does your heart work? Do you know what is your resting heart rate? Are you able to slow it down knowingly?  **Field outing:**  Use instructions of Arduino – pulse measuring software? | PC with Arduino software  Arduino UNO  and Heart Pulse Sensor |  | **In Classroom**  45 minutes |

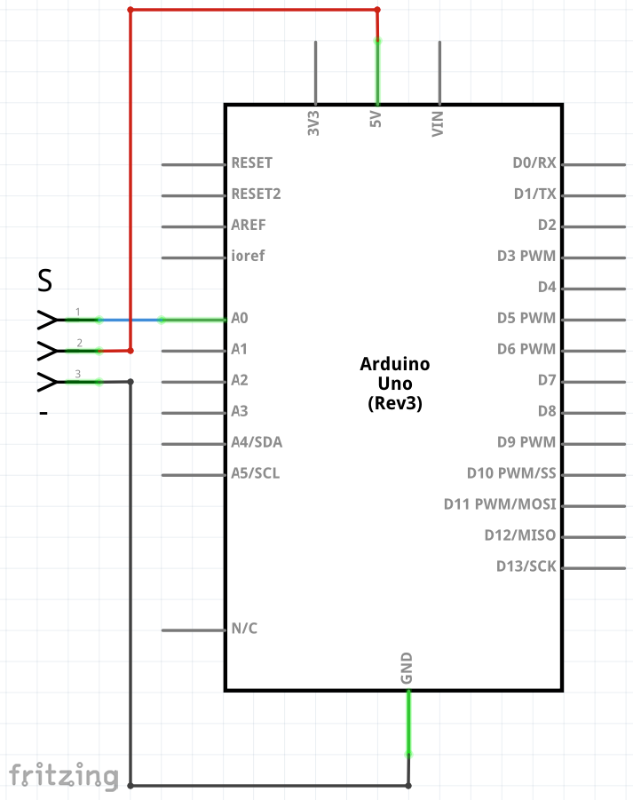
|  |  |
| --- | --- |
| **Description of the activity:** | |
| **Introduction** |  |
| **Main activity** | Heart Pulse Sensor is an input module that includes a combination of an infrared LED and a phototransistor. IF we insert a finger between the IR LED and the phototransistor, the infrared light from the LED will light up our finger and we will be able to measure the variable voltage through the phototransistor. Changes in the electrical voltage will be caused by the blood flowing in the finger, so we can measure the heart rate in BPM (beats per minute) with fairly good accuracy. |
| **Lesson Guide (Step by step)** | For the successful connection of the Heart Pulse Sensor with the Arduino board, it is necessary to connect 3 connecting pins. We connect the extreme pin S with the A0 pin, the middle pin with a 5 V Arduino pin and the extreme pin "-" with the Arduino ground. |
| **Exercises (2 or 3 levels of difficulty)** | To make the measurement more attractive, it is possible make one measuring in the resting phase, and the second one after 15 knee bends, and thus to find out how long it takes to slow the heart down again. A lot of students have smart watches or bracelets so the measured figures can be compared. |
| **Conclusion and Evaluation** |  |
| **Notes:** |  |

Bibliography

* <https://youtu.be/rba2NrzmEdU> (instructions on Youtube to connect the sensor)
* <https://www.youtube.com/watch?v=hrPuAhKpaNc> (instructions to compile)

Pictures:

1. How to connect Heart Pulse Sensor to Arduino UNO



2. Pulse Monitor Test Script

*// Pulse Monitor Test Script*

int sensorPin = 0;

double alpha = 0.75;

int period = 100;

double change = 0.0;

double minval = 0.0;

void setup ()

{

Serial.begin (9600);

}

void loop ()

{

static double oldValue = 0;

static double oldChange = 0;

int rawValue = analogRead (sensorPin);

double value = alpha \* oldValue + (1 - alpha) \* rawValue;

Serial.print (rawValue);

Serial.print (",");

Serial.println (value);

oldValue = value;

delay (period);

}

3. Sample code

// Arduino senzor tepu srdce KY039

// nastavení čísel propojovacích pinů

#**define** indikLED 13

#**define** analogPin A0

// vytvoření konstanty pro zpoždění při detekci

**const** **int** zpozdeniMereni = 60;

**void** setup() {

// komunikace po sériové lince rychlostí 9600 baud

Serial.begin(9600);

// inicializace analogového pinu jako vstup

// a digitálního pinu jako výstup

pinMode(analogPin, INPUT);

pinMode(indikLED, OUTPUT);

}

**void** loop()

{

// vytvoření dočasných proměnných pro uložení výsledků

**static** **int** uderyZaMinutu = 0;

**int** tepovaFrekvence = 0;

// kontrola detekce tepu srdce podprogramem detekceTepu

if (detekceTepu(analogPin, zpozdeniMereni)) {

// výpočet frekvence tepu srdce

tepovaFrekvence = 60000 / uderyZaMinutu;

// rozsvícení indikační LED diody při detekovaném

// měření tepu

digitalWrite(indikLED, HIGH);

// vytištění informace o naměřené tepové frekvenci

if (tepovaFrekvence > 50 & tepovaFrekvence < 200) {

Serial.print("Tepova frekvence: ");

Serial.print(tepovaFrekvence);

Serial.println(" uderu za minutu (BPM).");

}

// vynulování proměnné pro další měření

uderyZaMinutu = 0;

} else {

// v případě žádné detekce tepu vypni indikační LED diodu

digitalWrite(indikLED, LOW);

}

// pauza programu do dalšího měření

delay(zpozdeniMereni);

// připočtení zpoždění pro další měření

uderyZaMinutu += zpozdeniMereni;

}

// podprogram pro detekci tepu a výpočet jeho frekvence

**bool** detekceTepu(**int** senzorPin, **int** zpozdeni) {

// vytvoření pomocných proměnných

**static** **int** maxHodnota = 0;

**static** **bool** SpickovaHodnota = false;

**int** analogHodnota;

**bool** vysledek = false;

// načtení analogové hodnoty ze senzoru

analogHodnota = analogRead(senzorPin);

// přepočet analogové hodnoty pro další výpočty

analogHodnota \*= (1000 / zpozdeni);

// upravení maximální hodnoty

if (analogHodnota \* 4L < maxHodnota) {

maxHodnota = analogHodnota \* 0.8;

}

// detekce špičkové hodnoty

if (analogHodnota > maxHodnota - (1000 / zpozdeni)) {

// nastavení nového maxima při detekované špičce

if (analogHodnota > maxHodnota) {

maxHodnota = analogHodnota;

}

// nastavení platnosti výsledku, když

// nebyla detekována špička

if (SpickovaHodnota == false) {

vysledek = true;

}

SpickovaHodnota = true;

} else if (analogHodnota < maxHodnota - (3000 / zpozdeni)) {

SpickovaHodnota = false;

// upravení maximální hodnoty při změně měřených hodnot

maxHodnota -= (1000 / zpozdeni);

}

// vrácení výsledku podprogramu

return vysledek;

}