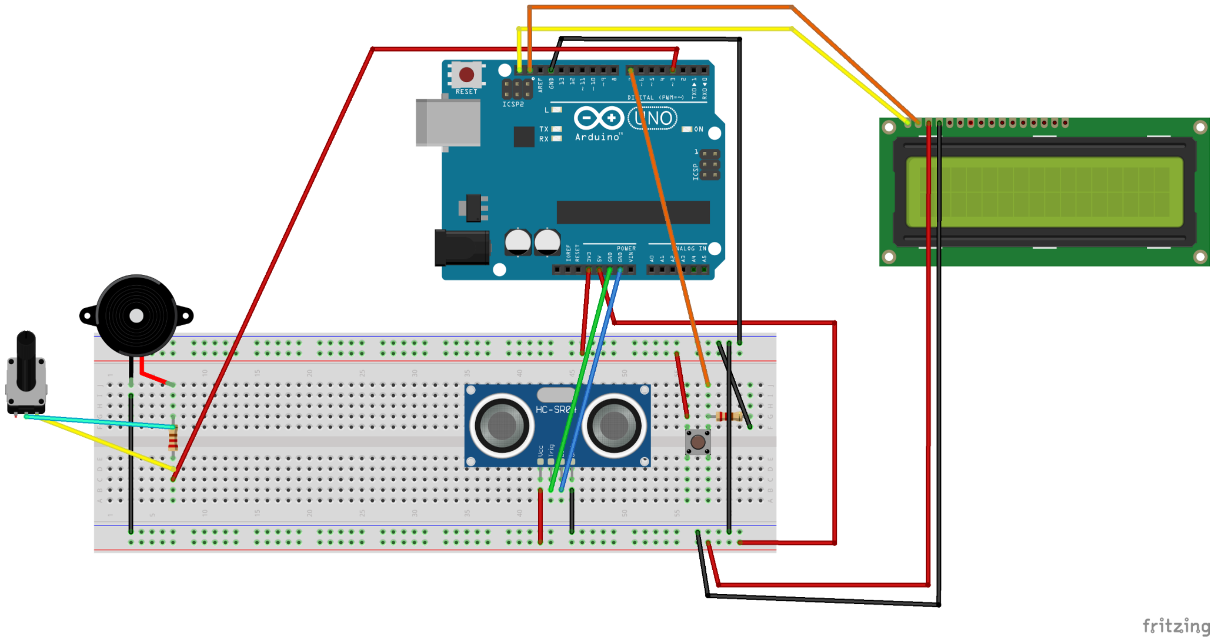
Final project PPC

Hielke Veenstra s2768887 & Marije Rekker s2634678

For this project we did not know what to make. We needed to find a solution to a day to day problem. So, what is it that we as students struggle with the most… Finding the time and motivation to actually study when there are always fun things to do when you're living on campus. So how do you get to work and make good use of your time? Well, it's scientifically proven that the pomodoro technique is the way to go!

How cool would it be to have a fancy physical pomodoro timer, that is fun to use and laying around on your desk desperately waiting to be used. That's why we decided to make a pomodoro timer for this project using an arduino, an LCD screen, a lot of cool sensors and connect it to a processing program on our laptop.



Our system works as follows; you can start the timer on the laptop in processing by selecting the time you want to study. In this screen you can also stay up to date on the current time. When the timer runs out, you hear an explosion on the laptop while an alarm goes off on the arduino. This alarm can be stopped by hovering your hand past the distance sensor, or with the fun little button on the breadboard. You can also change the volume of the alarm by turning the potentiometer (counter) clockwise. On the LCD screen you can see how many seconds you're already studying, while on the laptop screen you can see time running out. When this happens you see a tomato explode which indicates that you have studied enough and you need a break :)

To reset the timer simply press reset and choose your preferred time for your next study session.

ARDUINO CODE

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define trigPin 14
#define echoPin 15
#define buzzer 2 //buzzer to arduino pin 9
#define buttonPin 5

LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line
display

int frequenty = 100;
int buttonstate = 0; //turns alarm on
int buttonOutput = 1;
int myTime;
long duration;
int distance;

void setup() {
  Serial.begin(9600); // begin serial communication with a baud rate of 9600
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output
  lcd.init();                      // initialize the lcd
  lcd.backlight();
}

void loop() {
  myTime = millis() / 1000;
  lcd.clear();
  lcd.print("Seconds: ");
  lcd.setCursor(10, 1);
  lcd.print(myTime);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  delay(50);
  if (Serial.read() == 1) {
    buttonstate = 1;
  }
  if (distance <= 20) {
    buttonstate = 0;
  }
  buttonOutput = digitalRead(buttonPin);
  if (buttonOutput == 1 && buttonstate == 0) {
    buttonstate = 1;
    delay(50);
  }
  else if (buttonOutput == 1 && buttonstate == 1) {
    buttonstate = 0;
    delay(50);
  }
  if (buttonstate == 1) {
    if (frequenty < 1000) {
      tone(buzzer, frequenty);
      delay(50);
      noTone(buzzer);
      delay(50);
      frequenty = frequenty + 30;
    }
    else {
      frequenty = 100;
    }
  }
}
```

PROCESSING CODE

```
import processing.serial.*;
import processing.sound.*;

boolean alarmOn=false;
PImage tomato;
SoundFile explosion;
Buttons button;
Serial port;
Timer t;
int startHours;
int startMinutes;
int startSeconds;

int phase = 0;

void setup() {
  size(1000, 800);
  port = new Serial(this, Serial.list() [0], 9600);
  port.bufferUntil('\n');
  button = new Buttons();
  t = new Timer();
  //startHours = hour();
  //startMinutes = minute();
  //startSeconds = second();
   explosion = new SoundFile(this, "boomboom.mp3");
   tomato = loadImage("tomato-revue-starlight.gif");
}


void draw() {
  //int passedMinutes = millis()/60000;
  //int passedSeconds = millis()/1000;
  image(tomato,0,0,width,height);
  float hours, minutes, seconds;
  hours = hour();
  minutes = minute();
  seconds = second();
  button.display();
  button.update(mouseX,mouseY);
  //int timerHere = Integer.parseInt(button.timer);

  if (phase == 0) {
    textSize(48);
    fill(125, 125, 125);
    textMode(CENTER);
    text("Current time: " + (int)hours + ":" + (int)minutes + ":" + (int)seconds,1,49);
    text("Time: " + (button.timer-t.minute()) + ":" + (60-t.second()), 1, height/2);
  }
  if (button.timer-t.minute()<0 && alarmOn==false){
    port.write(1);
    explosion.play();
    alarmOn=true;
  }
}

class Buttons {
  int blockWidth = 100;
  int blockHeight = 80;
  int timer = 0;
  void display() {
    rectMode(CENTER);
    rect(width/5, height/4, blockWidth , blockHeight);
    text("5",width/5+blockWidth/2,height/4);
    rect(width/5*2, height/4, blockWidth, blockHeight);
    text("15",width/5*2+blockWidth/2,height/4);
    rect(width/5*3, height/4, blockWidth, blockHeight);
    text("30",width/5*3+blockWidth/2,height/4);
    rect(width/5*4, height/4, blockWidth, blockHeight);
    text("start",width/5*4+blockWidth/2,height/4);
    rect(width/5*4, height/4*2, blockWidth, blockHeight);
    text("stop", width/5*4+blockWidth/2,height/4*2);
  }

  void update(float pointerX, float pointerY) {
    if (pointerX >= width/5-(blockWidth/2) && pointerX <= width/5+(blockWidth/2) &&
      pointerY >= height/4-(blockHeight/2) && pointerY <= height/4+(blockHeight/2) &&
mousePressed == true) {
        timer = 0;
    }
    if (pointerX >= width/5*2-(blockWidth/2) && pointerX <= width/5*2+(blockWidth/2) &&
      pointerY >= height/4-(blockHeight/2) && pointerY <= height/4+(blockHeight/2) &&
mousePressed == true) {
        timer = 14;
    }
    if (pointerX >= width/5*3-(blockWidth/2) && pointerX <= width/5*3+(blockWidth/2) &&
      pointerY >= height/4-(blockHeight/2) && pointerY <= height/4+(blockHeight/2) &&
mousePressed == true) {
        timer = 29;
    }
    if (pointerX >= width/5*4-(blockWidth/2) && pointerX <= width/5*4+(blockWidth/2) &&
      pointerY >= height/4-(blockHeight/2) && pointerY <= height/4+(blockHeight/2) &&
mousePressed == true) {
        t.start();
    }
    if (pointerX >= width/5*4-(blockWidth/2) && pointerX <= width/5*4+(blockWidth/2) &&
      pointerY >= height/4*2-(blockHeight/2) && pointerY <= height/4*2+(blockHeight/2) &&
mousePressed == true) {
        t.stop();
    }
  }
}

class Timer {
  int startTime=0;
  int stopTime=0;
  boolean active=false;

  void start() {
    startTime = millis();
    active=true;
  }
  void stop(){
    stopTime= millis();
    active=false;
  }
  int enlapsed(){
    int enlapsed;
    if(active){
      enlapsed = (millis()-startTime);
    }
    else {
      enlapsed = (stopTime - startTime);
    }
    return enlapsed;
  }
  int second(){
    return (enlapsed()/1000)%60;
  }
  int minute(){
    return (enlapsed()/(1000*60))%60;
  }
  int hour(){
    return (enlapsed()/(1000*60*60))%24;
  }
}
```