PBD **Name:** _____
Spring 2022 By signing my name above,
Final Exam I agree to the University of Ljubljana
06/06/2022 academic honesty policies.
Time Limit: 105 Minutes

This exam contains 4 pages and 5 questions. Total of points is 50.

---

1. (12 points) You are programming a microcontroller of a hat shown in Figure 1. The hat features a built-in accelerometer, a fan with adjustable rotation speed, and an LED light, all connected to an ArduinoUNO-like microcontroller board. The hat should automatically infer a user's physical activity and adjust the fan rotation speed accordingly.

   (a) (4 points) Write the part of the programme that reads fifty (50) consecutive accelerometer samples (reading all three axes) with 20 millisecond delay, then calculates **the acceleration mean intensity** ($\overline{a} = \frac{1}{N}\sum_{i=1}^{N}(a_i)$), where a_i is the acc. intensity at time i, **the standard deviation of the intensity** ($s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a_i - \overline{a})^2}$), and **the mean crossing rate** of acceleration intensity ($MCR = \frac{1}{N-1}\sum_{i=2}^{N}(I(sign(a_{i-1} - \overline{a}) * sign(a_i - \overline{a}) < 0))$), where $I(x)$ is the indicator function, equal to 1 if $x = true$, zero otherwise, and stores them in variables `accMean`, `accStdDev`, `accMCR`, respectively.
   The programme should take a 5 second break from sensing after the variables are calculated and then repeat the sensing cycle.

   The connected accelerometer is a three-axes analog accelerometer and its x, y, and z acceleration outputs are connected to Arduino pins A0, A1, A2, respectively. The values you read from these pins should be used as they are (no conversion needed).

   (b) (4 points) Write the part of the programme that uses the extracted features to infer a user's activity and adjust the fan speed.

   Once 50 samples are recorded and processed, the programme should take the `accMean`, `accStdDev`, `accMCR` values, and use `int inferPhysicalActivity(int mean, int stddev, int mcr);` function of an already included library (`<mltoolkit.h>`). This function returns 0 if a user is still, 1 if a user is walking, 2 if a user is running.

   The fan's rotation speed is proportional to the voltage supplied at its input pin. The fastest speed is achieved when 5V are at the input and then it linearly falls down as the voltage is decreased. The fan supports PWM input mode, and the Arduino board can output PWM on its pin 3 (connected to the fan) with ranges between 0 (always off) and 255 (always on). Write the code, so that the fan rotates with a quarter of the maximum speed if a user is still, half of the full speed if a user is walking, and with the full speed if a user is running.

   (c) (4 points) Write the code that turns the LED light (connected to pin 2) on when a user presses the button (connected to pin 4). The LED should lite up immediately, taking precedence over any other calculation. [hint: use interrupts]

   Note: you can write your code in the sketch below, or on a separate piece of paper, but clearly stating which part of the sketch your code should go to.

```
#include<mltoolkit.h>

int accMean;
int accStdDev;
int accMCR;

// Define other variables and functions as needed
int accIntensity;
int accXvalues[50];
int accYvalues[50];
int accZvalues[50];
int accIntensityValues[50];
int counter = 0;

int pinAccX = A0;
int pinAccY = A1;
int pinAccZ = A2;
int pinFan = 3;
int pinLED = 2;
int pinButton = 4;

void setup(){
  // define pin modes as needed

  pinMode(pinAccX, INPUT);
  pinMode(pinAccY, INPUT);
  pinMode(pinAccZ, INPUT);
  pinMode(pinFan, OUTPUT);
  pinMode(pinLED, OUTPUT);
  pinMode(pinButton, INPUT);

  // your code for part c)
  attachInterrupt(digitalPinToInterrupt(pinButton), liteUp, RISING);
}

void loop(){
  // your code

 if (counter < 50) {
     accXvalues[counter] = analogRead(pinAccX);
     accYvalues[counter] = analogRead(pinAccY);
     accZvalues[counter] = analogRead(pinAccZ);
     delay(20);
  } else {
     // calculating mean intensity
     for(int i=0; i<counter; i++){
         accIntensity = sqrt(accXvalues[i]*accXvalues[i] + accYvalues[i]*accYvalues[i] +
              accZvalues[i]*accZvalues[i]);
         accIntensityValues[i] = accIntensity;
         accMean += accIntensity;
     }
     accMean /= counter;

     // calculating standard deviation and MCR
     for(int i=0; i<counter; i++){
       accStdDev += pow((accIntensityValues[i] - accMean),2);
       if (i>0) {
           if((accIntensityValues[i] - accMean)*(accIntensityValues[i-1] - accMean) < 0)
               {
               accMCR++;
           }
       }
     }
     accStdDev = sqrt(accStdDev/counter);
     accMCR /= (counter-1);
     counter = 0;
     accMean = 0;
     accStdDev = 0;
```

```
67        accMCR = 0;
68
69        // your code - for part b)
70        switch(inferPhysicalActivity(accMean, accStdDev, accMCR)){
71              case 0:
72                        analogWrite(pinFan, 85);
73                        break;
74              case 1:
75                        analogWrite(pinFan, 170);
76                        break;
77              case 2:
78                        analogWrite(pinFan, 255);
79                        break;
80        }
81        delay(2000);
82    }
83
84  }
85
86    // your code - for part c)
87  void liteUp() {
88         digitalWrite(pinLED, HIGH);
89  }
```

Figure 1: Hat with a fan and LED light.

2. (9 points) Embedded systems.
"Failures we have in today's technologies could be disastrous in embedded networked systems" warn Estrin et al. in 2001. Argue for or against this statement. List at least three specific issues in your argument.

   See Exam Preparation slides on Ucilnica.

3. (10 points) Write True or False next to the following claims about background processing.
   (a) (2 points) Service can be bound to another component through onBind call. T
   (b) (2 points) A Foreground Service has to show a notification icon in the notification bar. T
   (c) (2 points) Coroutines launched within GlobalScope will be terminated if a user rotates the screen with the app opened. F
   (d) (2 points) Alarms instantiated through AlarmManager are removed once a user closes the app. F
   (e) (2 points) Once done with the given task, a Service can send the results back to the UI thread via onPostExecute callback. F

4. (6 points) Web platform.
Define and in one sentence describe each of the following: IaaS, PaaS, SaaS.

   IaaS – Infrastructure as a Service – A cloud computing solution that virtualizes access to the hardware allowing customer applications to scale according to potentially time-varying needs (example: Amazon Web Service).

   PaaS – Platform as a Service – A cloud computing solution that virtualizes a whole platform (e.g. MEAN stack and the hardware) allowing customer applications to scale according to potentially time-varying needs (example: Google App Engine).

   SaaS – Software as a Service – A solution where the whole end-user-facing software solution lives in the cloud. (example: MS Office 365).

5. (13 points) You are designing an Android app for trading used items (something like Bolha in Slovenia, Craigslist in the US, etc.). The app should get data about available items on the market from a remote REST server. The server sends data in the JSON format.
   (a) (3 points) UI: Describe how you would implement the app so that it has two views - one for showing a gallery view of the available items, and the other (opened once a user clicks on a selected item) showing item details (e.g. title, image, location, description, price, etc.). Note: define how transitioning from one view to another is implemented.

   The app should contain one Activity with a NavHostFragment element and two Fragments. The first one – ItemsListFragment – would contain a RecyclerView with elements (e.g. CardViews) that would show the item listings. An Adapter would be used to connect the views with data objects. The second – DetailsFragment –

would contain the necessary TextViews and ImageViews for showing the details of a selected item. The navigation from the ItemsListFragment to DetailsFragment is done via the navigation graph and defines the transition from one fragment to another and carries the ID of the selected listing as an attribute of the navigation action.

(b) (7 points) Data manipulation: Describe the data transfer/manipulation/storage elements of your app. Clearly define the purpose of each of the elements you use. The app should provide a caching functionality as well, i.e. the items a user has seen should be available for offline viewing. Focus on the browsing part only, i.e. you don't need to support a seller's view and uploading the descriptions.

The Fragments should obtain the data from ItemsViewModel. Within ItemsViewModel the data is stored in LiveData objects observed by the Fragments. The ViewModel communicates to Repository class. This class manages both querying of the remote REST API as well as storing/fetching from a local database. For the database we can use Room and write the data access object (DAO) defining the operations on the database, such as storing/fetching listings. One class we need to define is an @Entity object Item that defines the table used for storing item listings in the local database. For remote REST API querying we can use Retrofit and a GSON converter, and define the data transfer objects (DTOs) for (de)serializing the data received via the API. Once ran, the app pulls data from the server via the Repository, stores it in the database, and shows listings to the user via the Fragments that obtain the information from the ItemsViewModel.

(c) (3 points) Synchronization: The data on the server can change, for example, because a new item appears, the price changes, or an item gets sold (note: for now, assume that sold item still remains on the server, just with a field "sold" set to "true", i.e. it does not get deleted). Describe how you would ensure that the up-to-date data is fetched from the server once a day, if the phone is connected to a WiFi network and connected to a charger.

We can enqueue a 24-hour periodic WorkManager request constrained on RequiresCharging and NetworkType=UNMETERED. In the Worker's doWork method we initiate a call for fetching the data from the server via the Repository class.

Note: your solution should be as efficient as possible. **Please draw diagram(s) of your solution.**